

# Clone Structures in Voters' Preferences

Edith Elkind

School of Physical and  
Mathematical Sciences  
Nanyang Technological University  
Singapore

Piotr Faliszewski

Department of Computer Science  
AGH University of Science and  
Technology, Poland

Arkadii Slinko

Department of Mathematics  
University of Auckland  
New Zealand

October 19, 2011

## Abstract

In elections, a set of candidates ranked consecutively (though possibly in different order) by all voters is called a clone set, and its members are called clones. A clone structure is a family of all clone sets of a given election. In this paper we study properties of clone structures. In particular, we give an axiomatic characterization of clone structures, show their hierarchical structure, and analyze clone structures in single-peaked and single-crossing elections. We give a polynomial-time algorithm that finds a minimal collection of clones that need to be collapsed for an election to become single-peaked, and we show that this problem is NP-hard for single-crossing elections.

## 1 Introduction

Group decision making plays an important role in the proper functioning of human societies and multiagent systems. Collective decisions are often made by aggregating the preferences of individual agents by means of *voting*: each agent ranks the available alternatives, and a voting rule is used to select one or more winners (see [1] for a general overview of voting, and [12] for a more algorithmic perspective). In general, the structure of the set of alternatives may be quite complex. For instance, Ephrati and Rosenschein [10] explore the situation where multiple agents try to coordinate their actions in order to devise a global plan. There the space of alternatives, i.e., of possible plans, may be huge, with some alternatives being very similar to each other. In such a case it may be reasonable to establish which plans differ fundamentally, and which are viewed as minor variations of each other.

Such structured decision-making environments have been studied in the social choice literature: for instance, Laffond et al. [16] describe the situation when a group of agents has

to choose from a set that is partitioned into several “projects,” where each project is defined as a set of possible variants. In this setting, all agents are likely to rank the variants of each projects contiguously. This model was further investigated by Laslier [17, 18]. Tideman [24] suggests a different explanation of why several alternatives in an election may be very similar to each other: a malicious party may try to “duplicate” an existing candidate in order to change the voting outcome. This procedure is known as *cloning* and the alternatives that appear together in all preference orders (though not necessarily in the same order) are called *clones*. Elkind et al. [9] study algorithms for cloning and show that optimal cloning is easy for many voting rules.

Both when clones arise naturally and when they are created by a manipulator, it may be useful to understand the internal structure of the resulting clone sets. Indeed, such an understanding could be instrumental in uncovering hidden properties of voters’ preferences such as, for example, a hierarchical structure of the alternative set, or the fact that after collapsing a small number of clones the election becomes single-peaked or single-crossing (informally, both single-peaked [4] and single-crossing [19] elections model societies focused on a single issue, such as, e.g., taxation level). In either case we could run the election in a better way by using a more suitable voting rule: in the former case we can use hierarchic voting, and in the latter case we can use the median voter rule—which is known to be strategy-proof for single-peaked (single-crossing) profiles—to select a group of clones, and then pick the final winner among them. Such an approach is likely to produce a better voting outcome as well as reduce the voters’ incentives for manipulation.

Our goal in this paper is to provide a formal understanding of what families of clone sets—which we call *clone structures*—can arise in elections (we give an axiomatic characterization), to provide convenient means of representing them (we show that PQ-trees of Booth and Lueker [5] very conveniently describe clone structures), and to find a polynomial-time algorithms that restores single-peakedness/single-crossingness in elections by collapsing a minimal number of clones (we succeed for the case of single-peakedness and prove NP-hardness for the case of single-crossing). We believe that our results are useful for understanding the impact of clones in decision-making scenarios, and will help in developing algorithms for settings where some of the candidates may be very similar to each other. Due to space limit, almost all proofs, as well as some discussions, are relegated to the appendix.

## 2 Preliminaries

Given a finite set  $C$  of *candidates* (or *alternatives*), a *preference order* (or *ranking*) over  $C$  is a total order over  $C$ , i.e., a complete, transitive and antisymmetric relation on  $C$ . Intuitively, a preference order is a ranking of the candidates from the most desirable one to the least desirable one. By  $\succleftarrow$  we denote an order obtained by reversing order  $\succ$ , that is,  $j \succleftarrow i$  if and only if  $i \succ j$ . For two disjoint sets  $X, Y \subseteq C$  and an order  $\succ$ , we write  $X \succ Y$  if  $x \succ y$  for all  $x \in X$  and all  $y \in Y$ . Given two sets  $X, Y \subseteq C$ , we say that  $X$  is a *proper subset* of  $Y$  if  $X \subseteq Y$  and  $1 < |X| < |Y|$ . We say that  $X$  and  $Y$  *intersect non-trivially* and write  $X \bowtie Y$  if  $X \cap Y \neq \emptyset$ ,  $X \setminus Y \neq \emptyset$  and  $Y \setminus X \neq \emptyset$ .

A *preference profile*  $\mathcal{R} = (R_1, \dots, R_n)$  on  $C$  is a collection of  $n$  preference orders over  $C$ , where each order  $R_i$ ,  $1 \leq i \leq n$ , represents the preferences of the  $i$ -th voter; for readability, we sometimes write  $\succ_i$  in place of  $R_i$ . An *election* over  $C$  is a pair  $\mathcal{E} = (C, \mathcal{R})$ , where  $\mathcal{R}$  is a preference profile over  $C$ . A *voting rule* is a mapping  $\mathcal{F}$  that, given an election  $\mathcal{E}$  over  $C$ , outputs a set  $\mathcal{F}(\mathcal{E}) \subseteq C$ ; the elements of  $\mathcal{F}(\mathcal{E})$  are called the *election winners*. Many voting rules are used in practice and studied theoretically; see [1]. However, since we focus on the nature of preference profiles, our results do not depend on the choice of a voting rule.

**Example 2.1.**  $\mathcal{R} = (R_1, R_2, R_3)$  with  $R_1 : a \succ_1 b \succ_1 c \succ_1 d$ ,  $R_2 : b \succ_2 d \succ_2 c \succ_2 a$ ,  $R_3 : a \succ_3 b \succ_3 d \succ_3 c$  is a preference profile over  $C = \{a, b, c, d\}$ .

The following definition, inspired by [24], is fundamental for our work.

**Definition 2.2.** Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a preference profile over a candidate set  $C$ . We say that a non-empty subset  $X \subseteq C$  is a *clone set* for  $\mathcal{R}$  if for every  $c, c' \in X$ , every  $a \in C \setminus X$ , and every  $i = 1, 2, \dots, n$  it holds that  $c \succ_i a$  if and only if  $c' \succ_i a$ .

Unlike Tideman [24], we define singletons to be clone sets; in the election from Example 2.1 each of  $\{a\}$ ,  $\{b\}$ ,  $\{c\}$ ,  $\{d\}$ ,  $\{d, c\}$ ,  $\{b, c, d\}$ , and  $\{a, b, c, d\}$  is a clone set.

### 3 Axiomatic Characterization of Clone Structures

Our first goal is to understand which set families can be obtained as clone structures. That is, given a collection  $\mathcal{C}$  of subsets of a candidate set  $C$ , we want to determine if there exists a preference profile  $\mathcal{R}$  over  $C$  such that each clone in  $\mathcal{R}$  appears in our collection and vice versa; we will say that such  $\mathcal{R}$  *implements*  $\mathcal{C}$ . The main technical results of this section are (a) an axiomatic characterization of implementable collections of subsets, and (b) a polynomial-time algorithm for recognizing such families.

In this section, we will consider elections over the set  $[m] = \{1, \dots, m\}$ . We will write  $[j, k]$  to denote  $\{j, j+1, \dots, k\}$  for  $j, k \in [m]$ .

**Definition 3.1.** Given a profile  $\mathcal{R} = (R_1, \dots, R_n)$  over  $[m]$ , let  $\mathcal{C}(\mathcal{R}) \subseteq 2^{[m]}$  be the collection of all clone sets for  $\mathcal{R}$ . We say that a family  $\mathcal{C} \subseteq 2^{[m]}$  is a *clone structure* on  $[m]$  if it is equal to  $\mathcal{C}(\mathcal{R})$  for some profile  $\mathcal{R}$  on  $[m]$ .

We remark that clone structures are very close in spirit to clans in 2-structures [8], and many results in this section resemble those for clans; the characterization given in this section and high-level proof approach are close in spirit to those of Möhring [20]. We will, however, present a direct argument rather than translate these prior results, both because we need intermediate results for the analysis of single-peaked elections and because such a translation is non-trivial and would obscure useful intuition.

**Example 3.2.** Let  $\mathcal{R}$  consist of a single linear order  $R : 1 \succ 2 \succ \dots \succ m$ . Then  $\mathcal{C}(\mathcal{R}) = \{[i, j] \mid i \leq j\}$  (see Figure 1(a)). Let  $\mathcal{R}'$  be a cyclic profile on  $[m]$ , i.e.,  $\mathcal{R}' = (R_1, \dots, R_m)$ , and the preferences of the  $i$ -th voter are given by  $R_i : i \succ_i i+1 \succ_i \dots \succ_i m \succ_i 1 \succ_i \dots \succ_i i-1$ . Then  $\mathcal{C}(\mathcal{R}') = \{[m]\} \cup \{\{i\} \mid i \in [m]\}$  (see Figure 1(b)).

We call the first clone structure from Example 3.2 a *string of sausages* and the second one a *fat sausage*. Note that any clone structure over  $[m]$  consists of at most  $\frac{m(m+1)}{2}$  sets, since each clone set can be described by its location (i.e., beginning and end) in the preference ordering of a fixed voter. Thus, a string of sausages and a fat sausage can be thought of as, respectively, the maximal and the minimal clone structure over  $[m]$ . Let us now establish some basic properties of clone structures.

**Proposition 3.3.** *Let  $\mathcal{R}$  be a profile on  $[m]$ . Then (1)  $\{i\} \in \mathcal{C}(\mathcal{R})$  for any  $i \in [m]$ ; (2)  $\emptyset \notin \mathcal{C}(\mathcal{R})$  and  $[m] \in \mathcal{C}(\mathcal{R})$ ; (3) if  $C_1, C_2$  are in  $\mathcal{C}(\mathcal{R})$  and  $C_1 \cap C_2 \neq \emptyset$ , then  $C_1 \cup C_2$  and  $C_1 \cap C_2$  are in  $\mathcal{C}(\mathcal{R})$ ; (4) if  $C_1, C_2$  are in  $\mathcal{C}(\mathcal{R})$  and  $C_1 \bowtie C_2$ , then  $C_1 \setminus C_2$  and  $C_2 \setminus C_1$  are in  $\mathcal{C}(\mathcal{R})$ .*

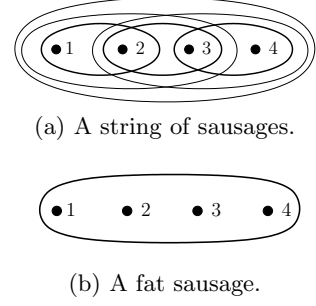


Figure 1: Diagrams representing clone structures from Example 3.2 for  $m = 4$ .

Proposition 3.3 does not give sufficient conditions for a family of subsets of  $[m]$  to be a clone structure. For example,  $P = 2^{[m]} \setminus \{\emptyset\}$ , where  $m \geq 3$ , satisfies all the conditions of Proposition 3.3. Yet, the cardinality of  $P$  is  $2^m - 1$ , whereas each clone structure over  $[m]$  has at most  $\frac{m(m+1)}{2}$  elements. The next proposition provides a further necessary condition for a family of subsets of  $[m]$  to be a clone structure. It is strong enough to exclude the collection  $2^{[m]} \setminus \{\emptyset\}$  for  $m > 3$ .

Given a profile  $\mathcal{R}$  over  $[m]$  and a set  $X \in \mathcal{C}(\mathcal{R})$ , we say that a set  $Z \in \mathcal{C}(\mathcal{R})$  is a *proper minimal superset* of  $X$  if  $X \subseteq Z$ ,  $X \neq Z$ , and there is no set  $Y \in \mathcal{C}(\mathcal{R})$  such that  $X \neq Y$ ,  $Y \neq Z$  and  $X \subseteq Y \subseteq Z$ .

**Proposition 3.4.** *For any profile  $\mathcal{R}$  on  $[m]$ , each  $X \in \mathcal{C}(\mathcal{R})$  has at most two proper minimal supersets in  $\mathcal{C}(\mathcal{R})$ .*

Note, however, that for  $m = 3$  the set family  $2^{[m]} \setminus \{\emptyset\}$  satisfies the conclusion of Proposition 3.4. Yet, it is obviously not a clone structure, since it contains a “cycle”  $\{1, 2\}, \{2, 3\}, \{3, 1\}$ . More generally, consider a set family over  $[m]$  that can be obtained from a string of sausages by adding the “missing link”, i.e., the set  $\{m, 1\}$  as well as all of its supersets that are necessary to satisfy the conclusions of Proposition 3.3; we will call this set family a *ring of sausages*. Clearly, a ring of sausages is not a clone structure, because it cannot be implemented by an acyclic preference relation; yet, the conclusion of Proposition 3.4 is satisfied. Thus, we need to forbid rings of sausages; in fact, we require a somewhat more general condition.

**Definition 3.5.** *We say that a set family  $\{A_0, \dots, A_{k-1}\}$  is a bicycle chain if  $k \geq 3$  and for all  $i = 0, \dots, k-1$  it holds that (1)  $A_{i-1} \bowtie A_i$ ; (2)  $A_{i-1} \cap A_i \cap A_{i+1} = \emptyset$ ; (3)  $A_i \subseteq A_{i-1} \cup A_{i+1}$ , where all indices are computed modulo  $k$ .*

**Proposition 3.6.** *If  $\mathcal{C}$  is a clone structure, it does not contain a bicycle chain.*

Propositions 3.3, 3.4 and 3.6 lead to the following set of axioms (note that these axioms are not normative; they simply tell us what clone structures *are* and not what they *should* be):

- A1.**  $\{f\} \in \mathcal{F}$  for any  $f \in F$ ,  $\emptyset \notin \mathcal{F}$ , and  $F \in \mathcal{F}$ .
- A2.** if  $C_1$  and  $C_2$  are in  $\mathcal{F}$  and  $C_1 \cap C_2 \neq \emptyset$ , then  $C_1 \cup C_2$  and  $C_1 \cap C_2$  are in  $\mathcal{F}$ .
- A3.** If  $C_1$  and  $C_2$  are in  $\mathcal{F}$  and  $C_1 \bowtie C_2$ , then  $C_1 \setminus C_2$  and  $C_2 \setminus C_1$  are in  $\mathcal{F}$ .
- A4.** Each  $C \in \mathcal{F}$  has at most two proper minimal supersets in  $\mathcal{F}$ .
- A5.**  $\mathcal{F}$  does not contain a bicycle chain.

Our next goal is to show that these five axioms indeed characterize clone structures. Axioms A1–A3 and axioms A4–A5 play different roles in our characterization result: the former ones ensure sufficient richness of a given set family, while the latter ones prevent it from being “too rich.”

We will first build up the necessary tools for our inductive argument. Let  $\mathcal{E}$  and  $\mathcal{F}$  be two families of subsets on two disjoint finite sets  $E$  and  $F$ , respectively. We can embed  $\mathcal{F}$  into  $\mathcal{E}$  as follows. Given  $e \in E$ , let  $\mathcal{E}(e \rightarrow \mathcal{F})$  denote the family of subsets  $\mathcal{E}' \cup \mathcal{F} \subseteq 2^{(E \setminus \{e\}) \cup F}$ , where  $\mathcal{E}'$  is obtained from  $\mathcal{E}$  by replacing each set  $X$  containing  $e$  with  $(X \setminus \{e\}) \cup F$ .

**Example 3.7.** Consider set families  $\mathcal{D} = \{\{x\}, \{y\}, \{x, y\}\}$  and  $\mathcal{C} = \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$  (both are strings of sausages and hence clone structures). Then,  $\mathcal{C}(b \rightarrow \mathcal{D}) = \{\{a\}, \{x\}, \{y\}, \{x, y\}, \{c\}, \{a, x, y\}, \{x, y, c\}, \{a, x, y, c\}\}$ . It is easy to check that this, again, is a clone structure.

If  $\mathcal{E}$  and  $\mathcal{F}$  satisfy axioms A1–A5 then so does  $\mathcal{E}(e \rightarrow \mathcal{F})$ . We prove it directly (it also follows from Theorem 3.12 combined with Proposition 3.11).

**Proposition 3.8.** Let  $\mathcal{E}$  and  $\mathcal{F}$  be families of subsets on disjoint sets  $E$  and  $F$ , respectively, that satisfy A1–A5. Then for any  $e \in E$  the set family  $\mathcal{E}(e \rightarrow \mathcal{F})$  also satisfies A1–A5.

Next, we define an inverse operation to embedding, which we call *collapsing*. Observe that when we embed  $\mathcal{F} \subseteq 2^F$  into  $\mathcal{E} \subseteq 2^E$ , any  $C \in \mathcal{E}(e \rightarrow \mathcal{F})$  is either a subset of  $F$ , a superset of  $F$ , or does not intersect  $F$  at all. Thus, for a set family  $\mathcal{C}$  on  $A$  to be collapsible, it should contain a set  $A'$  that does not intersect non-trivially with any other set in  $\mathcal{C}$ .

**Definition 3.9.** Let  $\mathcal{F}$  be a family of subsets on a finite set  $F$ . A subset  $\mathcal{E} \subseteq \mathcal{F}$  is a subfamily of  $\mathcal{F}$  if there is a set  $E \in \mathcal{F}$  such that (i)  $\mathcal{E} = \{F \in \mathcal{F} \mid F \subseteq E\}$ ; (ii) for any  $X \in \mathcal{F} \setminus \mathcal{E}$  we have either  $E \subseteq X$  or  $X \cap E = \emptyset$ . The set  $E$  is called the support of  $\mathcal{E}$ .  $\mathcal{E}$  is called a proper subfamily of  $\mathcal{F}$  if  $E$  is a proper subset of  $F$ .

One can check that if  $\mathcal{F}$  satisfies A1–A5 and  $\mathcal{E}$  is a subfamily of  $\mathcal{F}$ , then  $\mathcal{E}$  satisfies A1–A5 as well. Note that we require  $E \in \mathcal{F}$  (rather than just  $E \subseteq F$ ), and hence  $E \in \mathcal{E}$ .

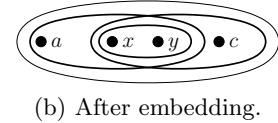
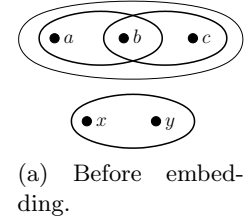


Figure 2: Clone structures from Example 3.7.

Let  $\mathcal{F}$  be a family of subsets on  $F$  that satisfies A1–A5 and let  $\mathcal{E}$  be a proper subfamily of  $\mathcal{F}$  on  $E \subset F$ . Then no set  $Y \in \mathcal{F}$  intersects  $E$  non-trivially, and hence  $\mathcal{E}$  can be “collapsed”. That is, we can obtain a new set family  $\mathcal{B}$  from  $\mathcal{F}$  by picking some alternative  $b \notin F$ , removing all sets  $X \in \mathcal{E} \setminus \{E\}$  from  $\mathcal{F}$ , and replacing each set  $Y$  that contains  $E$  with  $(Y \setminus E) \cup \{b\}$ . It is not hard to check that  $\mathcal{B}$  satisfies A1–A5; the proof is similar to that of Proposition 3.8. We will write  $\mathcal{F}(\mathcal{E} \rightarrow b)$  to denote the set family obtained by collapsing a subfamily  $\mathcal{E}$  of  $\mathcal{F}$ . That is, we have  $\mathcal{B} = \mathcal{F}(\mathcal{E} \rightarrow b)$  if and only if  $\mathcal{F} = \mathcal{B}(b \rightarrow \mathcal{E})$ .

Suppose that  $\mathcal{F}$  has no proper subfamilies; we will call such subset families *irreducible*.

**Theorem 3.10.** *Any irreducible family of subsets satisfying A1–A5 is either a string of sausages or a fat sausage.*

Thus, any irreducible set family that satisfies A1–A5 is a clone structure. This provides the basis for our inductive argument. For the inductive step, we need to show that if  $\mathcal{C}$  and  $\mathcal{D}$  are two clone structures over disjoint sets  $C$  and  $D$ , and  $c$  is some candidate in  $C$ , then  $\mathcal{C}(c \rightarrow \mathcal{D})$  is a clone structure. However, the proof of this fact is somewhat more complicated than one might expect. Indeed, suppose that we have a pair of profiles  $\mathcal{R} = (R_1, \dots, R_n)$  and  $\mathcal{Q} = (Q_1, \dots, Q_n)$  over sets  $C$  and  $D$ , respectively, such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$  and  $\mathcal{D} = \mathcal{C}(\mathcal{Q})$ . One might think that, given  $c \in C$ , we can obtain a preference profile  $\mathcal{R}'$  such that  $\mathcal{C}(c \rightarrow \mathcal{D}) = \mathcal{C}(\mathcal{R}')$  simply by substituting  $Q_i$  for  $c$  in  $R_i$ , for  $i = 1, \dots, n$ . This intuition is not entirely correct: without additional precautions, we may introduce “parasite” clones, i.e., clones that cross the boundary between  $C$  and  $D$ . However, we can construct  $\mathcal{R}'$ , containing  $n$  preference orders, from  $\mathcal{R}$  and  $\mathcal{Q}$  by tweaking this construction slightly.

**Proposition 3.11.** *Let  $\mathcal{C}$  and  $\mathcal{D}$  be two clone structures over sets  $C$  and  $D$ , respectively, where  $|C| = m$ ,  $|D| = k$ , and  $C \cap D = \emptyset$ . Then for each  $c \in C$ , the family of subsets  $\mathcal{C}(c \rightarrow \mathcal{D})$  is a clone structure.*

**Theorem 3.12.** *A family  $\mathcal{F}$  of subsets of  $[m]$  is a clone structure if and only if it satisfies conditions A1–A5.*

Based on Theorem 3.12, it is easy to test in polynomial time (and, in fact, even in logarithmic space) if a given set family (represented explicitly as a list of subsets) is a clone structure: one simply needs to check if all the axioms hold. We state this result more formally in Appendix A.1.

## 4 Compact Representations of Clone Structures

Let us now consider the issue of representing clone structures. We say that a clone structure  $\mathcal{C}$  is *k-implementable* if there is a *k*-voter profile  $\mathcal{R}$  such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ . One might expect that to obtain a complex clone structure we need an election with many voters. Yet, each clone structure can be implemented by a profile with at most three voters.

**Theorem 4.1.** *Any clone structure is 3-implementable.*

Nonetheless, we would like a more structured representation. In the previous section we have seen that clone structures are organized hierarchically and, thus, it is natural to represent them using trees. The specific type of trees that are most convenient for this task are PQ-trees introduced by Booth and Lueker [5].

A PQ-tree  $T$  over a set  $A = \{a_1, \dots, a_n\}$  is an ordered tree that represents a family of permutations over  $A$  as follows. The leaves of the tree correspond to the elements of  $A$ . Each internal node is either of type P or of type Q. A *frontier* of  $T$  is a permutation of  $A$  obtained by reading the leaves of  $T$  from left to right (recall that  $T$  is ordered). The following operations are allowed on the tree: If a node is of type P, then its children can be permuted arbitrarily. If a node is of type Q, then the order of its children can be reversed. A given permutation  $\pi$  of  $A$  is *consistent* with a PQ-tree  $T$ , if we can obtain  $\pi$  as the frontier of  $T$  by applying the above operations.

We now describe a natural way to represent clone structures as PQ-trees. Consider a clone structure  $\mathcal{C}$  over a finite set  $C$ . Our characterization of irreducible clone structures implies that any two proper irreducible subfamilies of  $\mathcal{C}$  have non-intersecting supports.

**Proposition 4.2.** *Let  $\mathcal{C}$  be a clone structure over a finite set  $C$ , and let  $\mathcal{B}$  and  $\mathcal{D}$  be two proper irreducible subfamilies of  $\mathcal{C}$  on sets  $B \subseteq C$  and  $D \subseteq C$ , respectively. Then  $B \cap D = \emptyset$ .*

Proposition 4.2 implies that every element of  $C$  belongs to at most one proper irreducible subfamily of  $\mathcal{C}$ . Thus, given a clone structure  $\mathcal{C} \subseteq 2^C$ , there is a unique maximal collection of pairwise disjoint sets  $\text{Dec}(\mathcal{C}) = \{C_1, \dots, C_k\}$  such that  $C_i \subseteq C$ ,  $|C_i| \geq 2$ , and for each  $i = 1, \dots, k$  the set family  $\mathcal{C}_i = \{C \in \mathcal{C} \mid C \subseteq C_i\}$  is an irreducible subfamily of  $\mathcal{C}$  (if  $\mathcal{C}$  is itself irreducible, then  $k = 1$  and  $C_1 = C$ ). This collection can be efficiently constructed by identifying the minimal (with respect to inclusion) non-singleton sets in  $\mathcal{C}$ : any such set of size  $s \geq 3$  is itself an irreducible clone structure (a fat sausage), and for a set of size  $s = 2$  we need to find the maximal string of sausages that contains it. Note that it need not be the case that  $\bigcup_{i=1}^k C_i = C$ ; some elements may not belong to any proper irreducible clone structure (consider, for instance, the clone structure over  $\{a, b, c, d\}$  given by  $\{\{a\}, \{b\}, \{c\}, \{d\}, \{b, c\}, \{a, b, c, d\}\}$ ). We will refer to the collection  $\text{Dec}(\mathcal{C})$  as the *decomposition* of  $\mathcal{C}$ .

We can now inductively define a PQ-tree  $T(\mathcal{C})$  associated with a clone structure  $\mathcal{C} \subseteq 2^C$  (for convenience, our PQ-trees will be labeled). Suppose first that  $\mathcal{C}$  is an irreducible clone structure over the set  $C = \{c_1, \dots, c_m\}$ . Then, by Theorem 3.10, it is either a string of sausages or a fat sausage. In the former case, assume without loss of generality that  $\mathcal{C}$  is associated with the order  $c_1 \succ c_2 \succ \dots \succ c_m$ , i.e., it contains sets  $\{c_i, c_{i+1}\}$  for  $i = 1, \dots, m-1$ . In both cases, we let  $T(\mathcal{C})$  to be a tree of depth 1 that has  $m$  (ordered) leaves. The  $i$ -th leaf is labeled by  $c_i$ . If  $\mathcal{C}$  is a string of sausages, the root of the tree is of type Q and is labeled by  $c_1 \oplus \dots \oplus c_m$ ; if  $\mathcal{C}$  is a fat sausage, the root is of type P and is labeled by  $c_1 \odot \dots \odot c_m$ . For  $m = 2$  the clone structure  $\mathcal{C}$  is both a string of sausages and a fat sausage; we treat it as a fat sausage.

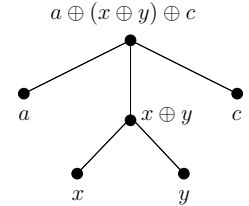


Figure 3: Tree representation of the embedded clone structure from Example 3.7.

Now, if  $\mathcal{C}$  is reducible, we compute its decomposition  $\text{Dec}(\mathcal{C}) = \{C_1, \dots, C_k\}$ . For  $i = 1, \dots, k$ , we set  $\mathcal{C}_i = \{X \in \mathcal{C} \mid X \subseteq C_i\}$ , pick  $c^1, \dots, c^k \notin C$ , and let  $\mathcal{C}'$  be the set family on the set  $C' = (C \setminus \bigcup_{i=1}^k C_i) \cup \{c^1, \dots, c^k\}$  given by  $\mathcal{C}' = \mathcal{C}(\mathcal{C}_1 \rightarrow c^1, \dots, \mathcal{C}_k \rightarrow c^k)$ . We then construct the tree  $T(\mathcal{C}')$ . This tree has leaves labeled by  $c^1, \dots, c^k$ . We replace each such leaf  $c^i$  by the labeled tree  $T(\mathcal{C}_i)$  for the irreducible set family  $\mathcal{C}_i$ .

Given the tree  $T(\mathcal{C})$ , we can reconstruct the clone structure  $\mathcal{C}$  in an obvious way. To illustrate this discussion, in Figure 3 we give a PQ-tree for the clone structure from Example 3.7. We remark that the descendants of any internal node of  $T(\mathcal{C})$  form a clone set. However, the converse is not necessarily true, i.e., there are clone sets that cannot be obtained in this way: if an internal node  $v$  is labeled with a string of sausages and has  $k$  children,  $k \geq 3$ , the descendants of any  $\ell$  consecutive children of  $v$ ,  $\ell < k$ , form a clone set. Indeed, it is not hard to see that any clone set corresponds either to a subtree of  $T(\mathcal{C})$  or to a collection of subtrees of  $T(\mathcal{C})$  whose roots are consecutive children of the same Q-node.

## 5 Clones in Single-Peaked Elections

It is not unusual for voters to make their decisions based on the candidates' position on a single prominent issue, such as, e.g., the level of taxation. Elections where all voters make their decisions in this way (with respect to the same issue) are known as *single-peaked*. In such elections, the candidates are ordered with respect to their position on the issue. This ordering is called *the societal axis*; it can, for example, order the candidates from those supporting the lowest level of taxation to those supporting the highest level of taxation. Each voter  $v$  forms her preference order as follows. First,  $v$  picks the candidate who is the closest to her ideal point on the societal axis. She then ranks the remaining candidates according to her perceived distance to the ideal point. The perception of distance may differ from one voter to another: some voters may view a large deviation to the right as less significant than a small deviation to the left, while others may hold the opposite view. Thus, the voter will zig-zag through the candidate list, ending her ranking with either the leftmost or the rightmost candidate. For example, if the possible tax rates are 10%, 15%, 20%, and 25%, and an election is single-peaked with respect to the axis  $10\% > 15\% > 20\% > 25\%$ , a voter's preference order may be, e.g.,  $20\% \succ 15\% \succ 10\% \succ 25\%$ , but not  $20\% > 10\% > 15\% > 25\%$ . Formally, a single-peaked preference profile is defined as follows.

**Definition 5.1.** Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a profile over a candidate set  $C$ , and let  $>$  be a linear order over  $C$  (the societal axis). We say that an order  $\succ$  over  $C$  is compatible with  $>$  if for all  $c, d, e \in C$  such that either  $c > d > e$  or  $e > d > c$ , it holds that  $c \succ d \implies d \succ e$ . We say that  $\mathcal{R}$  is single-peaked with respect to  $>$  if each preference order in  $\mathcal{R}$  is compatible with  $>$ . A profile  $\mathcal{R} \in \mathcal{L}^n(C)$  is called single-peaked if there exists a linear order over  $C$  such that  $\mathcal{R}$  is single-peaked with respect to  $>$ ; we say that  $>$  witnesses single-peakedness of  $\mathcal{R}$ .

The literature on single-peaked elections is vast; for examples and intuition we point the reader to the original paper of Black [4], which introduced this notion, and, for a more



algorithmic perspective, to some recent computational social choice papers [25, 11, 7, 6, 13].

There are several reasons why single-peaked elections received so much attention; the notion of single-peakedness is very natural, and indeed quite a few real-life elections are (close to) single-peaked. Further, single-peaked elections have many desirable properties, of which perhaps the best-known one is that they admit non-manipulable, non-dictatorial voting rules (e.g., the median voter rule [4]). This is a very attractive property, which distinguishes single-peaked elections from those with unrestricted preferences (see [15, 23]).

Unfortunately, if a candidate in a single-peaked election is cloned, the election may lose the single-peakedness property: if many voters find the clones very similar, they are likely to rank them randomly, which may be incompatible with any societal axis. Thus, given an election, we might want to check if it can be made single-peaked by “decloning.” Of course, we would like the resulting election to be close to the original one. Thus, we look for a single-peaked election that “collapses” as few clones as possible, i.e., has the maximum number of alternatives among all single-peaked elections that can be obtained from the original one by decloning. The main result of this section is a polynomial time algorithm for this problem. Our secondary goal is to understand which clone structures can arise in single-peaked elections. We give a partial answer by, on one hand, providing an example of a clone structure that cannot be implemented by a single-peaked profile, and, on the other hand, identifying a large family of clone structures that admit such an implementation.

**Preliminary Observations.** Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a preference profile over a candidate set  $C$ , and let  $D$  be some clone set in  $\mathcal{C}(\mathcal{R})$ . Given a  $c \notin C$ , we write  $\mathcal{R}(D \mapsto c)$  to denote the profile  $\mathcal{R}' = (R'_1, \dots, R'_n)$ , where each  $R'_i \in \mathcal{R}'$  is obtained from  $R_i$  by replacing the block of candidates from  $D$  with  $c$ . We refer to the process of converting  $\mathcal{R}$  to  $\mathcal{R}(D \mapsto c)$  as *decloning  $D$  to  $c$  in  $\mathcal{R}$* . Note that we can declone  $D$  even if the collection of subsets  $\mathcal{D} = \{X \in \mathcal{C}(\mathcal{R}) \mid X \subseteq D\}$  is not a subfamily of  $\mathcal{C}(\mathcal{R})$  (this is the only difference between decloning operation  $\mapsto$  and the collapsing operation  $\rightarrow$ ). Given a preference profile  $\mathcal{R}$  over a set of candidates  $C$ , we let  $c(\mathcal{R}) = |C|$ .

Given a profile  $\mathcal{R} = (R_1, \dots, R_n)$ , we refer to the top-ranked candidate in  $R_i \in \mathcal{R}$  as the *peak of  $R_i$* , and denote it by  $\text{peak}(R_i)$ . We write  $\text{peak}(\mathcal{R})$  to denote the set  $\{\text{peak}(R_i) \mid R_i \in \mathcal{R}\}$ . (Note that if  $\mathcal{R}$  is single-peaked, this does not imply that  $|\text{peak}(\mathcal{R})| = 1$ : intuitively, the term “single-peaked” refers to the shape of individual preference orders with respect to a given societal axis.) Theorem C.1 and surrounding discussion in the appendix clarifies the relation between possible societal axes and the set of peaks of a profile.

We are ready to start our investigation of clones in single-peaked profiles. Consider a preference profile  $\mathcal{R}$  over  $C$  that is single-peaked with respect to some order  $>$ , and let  $D$  be a clone set with respect to  $\mathcal{R}$ . Do members of  $D$  appear consecutively in  $>$ ? Not necessarily (take two votes,  $b \succ c \succ a \succ d$  and  $c \succ b \succ a \succ d$ , and axis  $a > b > c > d$ ;  $\{a, d\}$  is a clone even though  $a$  and  $d$  are not consecutive in  $>$ ), but they form at most two blocks within  $>$ .

**Proposition 5.2.** *Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a preference profile over a candidate set  $C$  that is single-peaked with respect to some order  $>$ , and let  $D \in \mathcal{C}(\mathcal{R})$  be a clone set for  $\mathcal{R}$  with  $|D| \geq 2$ . Then  $C$  can be partitioned into pairwise disjoint sets  $A_1, A_2, D_1, D_2$ , and  $P$  so that  $C \setminus D = A_1 \cup P \cup A_2$ ,  $D = D_1 \cup D_2$ ,  $D_1 \neq \emptyset$ ,  $D_2 \neq \emptyset$ , and  $A_1 > D_1 > P > D_2 > A_2$ .*

Further, if  $P \neq \emptyset$ , then  $\text{peak}(\mathcal{R}) \subseteq P$ , and, moreover,  $P \succ_i D \succ_i A_1 \cup A_2$  for each  $i = 1, \dots, n$ .

Proposition 5.2 motivates a very useful classification of clone sets in single-peaked profiles. Let  $\mathcal{R}$  be a profile over some candidate set  $C$  that is single-peaked with respect to some order  $>$ . If the elements of  $D$  are ranked contiguously in  $>$ , then we say that  $D$  is a *clone set of the first type* with respect to  $>$ ; otherwise, we say that  $D$  is a *clone set of the second type* with respect to  $>$ . The following is an immediate corollary of the second part of Proposition 5.2.

**Corollary 5.3.** *Let  $\mathcal{R}$  be a single-peaked preference profile. If  $D \in \mathcal{C}(\mathcal{R})$  is a clone set of the second type w.r.t. some societal axis  $>$ , then  $D$  does not contain any peaks of  $\mathcal{R}$ .*

Observe that if  $D$  is a clone set of the first type that does not contain any peaks of  $\mathcal{R}$ , then for each voter  $i$  either  $\text{peak}(R_i) \succ_i D$ , in which case  $\succ_i$  coincides with  $>$  on  $D$ , or  $D \succ_i \text{peak}(R_i)$ , in which case  $\succ_i$  coincides with  $\prec$  on  $D$ . Thus, the following corollary.

**Corollary 5.4.** *Let  $\mathcal{R}$  be a single-peaked preference profile over a candidate set  $C$ . If  $D \in \mathcal{C}(\mathcal{R})$  is a clone set of the first type with respect to some societal axis  $>$ ,  $|D| \geq 2$ , and  $D \cap \text{peak}(\mathcal{R}) = \emptyset$ , then  $D$  is a string of sausages.*

**Decloning Towards a Single-Peaked Profile.** We will now present an algorithm for transforming a given election into a single-peaked one by decloning. Our algorithm works with a PQ-tree  $T$  that captures the clone structure of our profile. Informally, it first contracts the tree to a single node, and then greedily reintroduces clone sets, following the branches of the tree, while maintaining the invariant that the resulting profile is single-peaked (this requires care as single-peakedness of a profile can be witnessed by many different societal axes). In what follows, we present a version of our algorithm that only declones clone sets that correspond to subtrees of  $T$ . This algorithm produces the optimal decloning for many settings (and, in particular, for all profiles whose PQ-trees contain P-nodes only), but may fail to find an optimal solution in some cases. We will show an example of such a case and outline a more sophisticated polynomial-time algorithm that works for all profiles. We describe our algorithm in terms of proper colorings of a PQ-tree.

**Definition 5.5.** *Let  $T$  be a PQ-tree. A coloring of  $T$  is a function  $f$  from the set of nodes of  $T$  to the set  $\{\text{black}, \text{white}\}$ . A coloring of  $T$  is proper if the children of each black node are black. Given a coloring  $f$  of  $T$ , let  $W(f)$  denote the set of nodes that are white under  $f$ .*

**Definition 5.6.** *Let  $\mathcal{R}$  be a profile over a candidate set  $C$ , set  $T = T(\mathcal{C}(\mathcal{R}))$ , and let  $f$  be a proper coloring of  $T$ . For a node  $v \in T$ , let  $C_v = \{c \in C \mid c \text{ is a leaf of } T\text{'s subtree rooted in } v\}$ . Define  $\mathcal{R}(T, f)$  to be the profile obtained from  $\mathcal{R}$  as follows: for each internal node  $v$ , if  $v$  is black and its parent is white (or  $v$  is the root), declone the set  $C_v$  to a single candidate  $v$ .*

Our algorithm BASICDECLONESP takes as an input a preference profile  $\mathcal{R}$ . It then constructs a PQ-tree  $T = T(\mathcal{C}(\mathcal{R}))$  that corresponds to the clone structure of  $\mathcal{R}$ . It initializes  $f$  to be a coloring of  $T$  in which every node is *black*; this coloring will be modified during the execution of the algorithm. Note that at this point  $\mathcal{R}(T, f)$  is single-peaked and  $c(\mathcal{R}(T, f)) = 1$ . The algorithm also maintains a queue of nodes that it intends to visit. Initially, the queue contains the root of  $T$  only. Throughout the execution, we ensure that  $\mathcal{R}(T, f)$  is single-peaked and all ancestors of each node in the queue are white; note that this is indeed the case just after the initialization stage.

At each stage, BASICDECLONESP( $\mathcal{R}$ ) picks a node  $v$  from the queue (if the queue is empty, the algorithm terminates). It then executes the following steps:

1. Set  $f(v) = \textit{white}$ .
2. Check if  $\mathcal{R}(T, f)$  is single-peaked (possible in polynomial time [3, 11]).
3. If  $\mathcal{R}(T, f)$  is single-peaked, add all children of  $v$  to the queue. Otherwise, reset  $f(v) = \textit{black}$ .

By induction on the execution of the algorithm, one can see that at each point in time  $f$  is a proper coloring of  $T$ , and therefore  $\mathcal{R}(T, f)$  is well-defined. Further, each node is processed at most once, so BASICDECLONESP( $\mathcal{R}$ ) runs in polynomial time. Finally, it is clear that BASICDECLONESP( $\mathcal{R}$ ) produces a single-peaked election.

We now show that given a profile  $\mathcal{R}$  over a candidate set  $C$ , BASICDECLONESP outputs a single-peaked profile  $\mathcal{R}'$  with the following property:  $c(\mathcal{R}') \leq c(\mathcal{R}'')$  for any single-peaked preference profile  $\mathcal{R}''$  that can be obtained from  $\mathcal{R}$  by decloning clone sets that correspond to subtrees of  $T(\mathcal{C}(\mathcal{R}))$ . First, as a sanity check, we note (Proposition C.2 in the appendix) that if a profile is already single-peaked then it remains single-peaked after decloning. Second, we show that the greedy way in which BASICDECLONESP reintroduces clones cannot prevent us from finding an optimal solution.

**Proposition 5.7.** *Let  $\mathcal{R}$  be a preference profile over a set of candidates  $C$ . Let  $D^1, \dots, D^k \in \mathcal{C}(\mathcal{R})$  be a sequence of pairwise disjoint clone sets, and let  $c^1, \dots, c^k$  be a sequence of distinct candidates not in  $C$ . For each  $i = 1, \dots, k$ , let  $\mathcal{R}_i$  denote a preference profile in which for each  $j = 1, \dots, k$ ,  $j \neq i$ ,  $D^j$  is decloned to  $c^j$ . Then  $\mathcal{R}$  is single-peaked if and only if each of the profiles  $\mathcal{R}_i$ ,  $i = 1, \dots, k$ , is single-peaked.*

The idea of the proof is to show that if reintroducing a given clone does not break single-peakedness of a profile, then one can obtain a societal axis witnessing this fact by local modifications of the societal axis used prior to the clone's reintroduction. In the next theorem we show correctness of BASICDECLONESP for the case of decloning subtrees only, and in the following proposition we show that sometimes decloning subtrees does not suffice.

**Theorem 5.8.** *Given a preference profile  $\mathcal{R}$  over a candidate set  $C$ , BASICDECLONESP( $\mathcal{R}$ ) runs in time polynomial in  $|\mathcal{R}|$  and  $|C|$ , and produces a single-peaked preference profile  $\mathcal{R}'$  such that  $c(\mathcal{R}') \leq c(\mathcal{R}'')$  for any single-peaked preference profile  $\mathcal{R}''$  that can be obtained from  $\mathcal{R}$  by decloning one or more clone sets that correspond to subtrees of  $T(\mathcal{C}(\mathcal{R}))$ .*

**Proposition 5.9.** *Let  $\mathcal{C}$  be a string of sausages over candidates  $\{a, b, c\}$ , let  $\mathcal{D}'$  be a string of sausages over candidates  $\{1, 2, 3\}$ , and let  $\mathcal{D}''$  be a fat sausage over candidates  $\{x, y\}$ . Then the clone structure  $\mathcal{C}' = \mathcal{C}(b \rightarrow \mathcal{D}')$  is not single-peaked, but the clone structure  $\mathcal{C}'' = \mathcal{C}(b \rightarrow \mathcal{D}'')$  is single-peaked.*

In Proposition 5.9 the clone structure  $\mathcal{C}''$  can be obtained from  $\mathcal{C}'$  by decloning the clone  $\{1, 2\}$  to  $x$  (and renaming 3 as  $y$ ). Further,  $\mathcal{C}''$  has four candidates, whereas any clone structure that can be obtained from  $\mathcal{C}'$  by decloning clone sets that correspond to substructures of  $\mathcal{C}$  will have at most three candidates. This shows that if  $T(\mathcal{C}(\mathcal{R}))$  contains Q-nodes, BASICDECLONESP( $\mathcal{R}$ ) may fail to find the optimal decloning of a given profile into a single-peaked one (note, however, that this issue does not arise if  $T(\mathcal{C}(\mathcal{R}))$  contains P-nodes only, in which case BASICDECLONESP( $\mathcal{R}$ ) actually finds an optimal solution).

Hence, to obtain an algorithm that *always* finds an optimal decloning, we need to modify BASICDECLONESP( $\mathcal{R}$ ) to also consider clone sets that correspond to substrings of strings of sausages. However, a straightforward implementation of this idea leads to an exponential blow-up in the running time: there are exponentially many ways to choose a non-overlapping collection of substrings of a given string. Fortunately, it turns out that it suffices to consider breaking strings of sausages into two parts. In Section D of the appendix we present a polynomial-time algorithm that makes use of this idea and always finds an optimal single-peaked decloning of a given preference profile; this algorithm differs from BASICDECLONESP( $\mathcal{R}$ ) in its handling of Q-nodes only.

**Clone Structures via Single-Peaked Profiles.** Let us now turn to our second goal, the problem of characterizing clone structures that can be implemented using single-peaked profiles. Formally, we say that a clone structure  $\mathcal{C}$  is *single-peaked* if there exists a single-peaked profile  $\mathcal{R}$  such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ .

**Proposition 5.10.** *Fat sausages and strings of sausages are single-peaked.*

Thus, by Proposition 5.9, single-peaked clone structures are not closed under embeddings. Nonetheless, we identify a large class of single-peaked clone structures.

**Proposition 5.11.** *Let  $\mathcal{C}$  be a clone structure over a set of candidates  $C$ . Suppose that for each Q-node of the PQ-tree decomposition  $T(\mathcal{C})$  of  $\mathcal{C}$  it holds that all children, except possibly the leftmost child and the rightmost child, are labeled with singletons, i.e., elements of  $C$ . Then  $\mathcal{C}$  is a single-peaked clone structure.*

The second part of Proposition 5.9 shows that Proposition 5.11 does not characterize single-peaked clone structures; finding such a characterization is an interesting open problem.

## 6 Clones in Single-Crossing Elections

Let us now consider a different domain restriction called *single-crossing*. The idea behind single-crossing elections is similar to that behind single-peaked elections, but now it is

the voters who are ordered along some axis; say, the traditional left-to-right spectrum of political views. Consider a voter  $v$  on one of the extreme ends of the spectrum and two candidates,  $c$  and  $d$  such that  $v$  prefers  $c$  to  $d$ . As we move toward the other end of the voter spectrum, for a while voters agree that  $c$  is better than  $d$ , but eventually  $d$  *crosses*  $c$  and, from this point on, the voters prefer  $d$  to  $c$ . Single-crossing dates back at least to Mirrlees [19]; see also [14, 22, 2] for more recent work that also describes realistic settings where single-crossing profiles arise. Formally, we use the following definition.

**Definition 6.1.** *We say that a preference profile  $\mathcal{R} = (R_1, \dots, R_n)$  over candidate set  $C$  is single-crossing with respect to order  $\triangleleft$  over  $[n]$ , if for every pair of distinct candidates  $c, d \in C$  it holds that either  $\{i \mid c \succ_i d\} \triangleleft \{j \mid d \succ_j c\}$  or  $\{i \mid d \succ_i c\} \triangleleft \{j \mid c \succ_j d\}$ . We say that a profile is single-crossing if there exists an order  $\triangleleft$  with respect to which it is single-crossing.*

Strictly speaking, the notion introduced in Definition 6.1 is referred to as *order restriction* and not single-crossing. However, in our setting these two notions are equivalent and the term “single-crossing” much more intuitively describes the notion.

We observe that one can check in polynomial time whether a given profile is single-crossing; to the best of our knowledge, this observation does not appear in the literature.

**Proposition 6.2.** *The problem of checking if a given profile is single-crossing is in P.*

As in the case of single-peakedness, we would like to know which clone structures can be implemented by single-crossing profiles and what is the complexity of decloning towards a single-crossing profile. We reach opposite answers from those for the case of single-peakedness.

**Theorem 6.3.** *For every clone structure  $\mathcal{C}$  there exists a single-crossing profile  $\mathcal{R}$  such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ .*

We remark that, unlike the construction in the proof of Proposition 3.11, which leads to a 3-voter profile (Theorem 4.1), the proof of Theorem 6.3 produces a profile with many voters.

**Theorem 6.4.** *Given a profile  $\mathcal{R}$  over a candidate set  $C$  and a positive integer  $k$ , it is NP-hard to decide if there exists a single-crossing profile  $\mathcal{R}'$  with  $c(\mathcal{R}') \geq k$  that can be obtained from  $\mathcal{R}$  by decloning.*

However, optimal decloning is easy if the order of voters is fixed.

**Proposition 6.5.** *Given a profile  $\mathcal{R}$  over a candidate set  $C$ , a positive integer  $k$ , and an order  $\triangleleft$ , we can decide in polynomial time if there exists a profile  $\mathcal{R}'$  with  $c(\mathcal{R}') \geq k$  that is single-crossing with respect to  $\triangleleft$  and can be obtained from  $\mathcal{R}$  by decloning.*

## 7 Conclusions and Future Work

We have characterized clone structures in elections, obtained a convenient representation using PQ-trees, and used this representation in an algorithm that restores an election’s single-peakedness by decloning as few candidates as possible. On the other hand, we have shown that recovering the single-crossing property optimally is NP-hard. We also made first steps toward characterizing clone structures in single-peaked elections and we have shown that all clone structures can be implemented with single-crossing profiles.

Other research directions include establishing the complexity of verifying whether a given candidate can be made an election winner (under a particular voting rule) by decloning a given number of candidates. We hope that our work will facilitate obtaining hardness results for this problem, thus complementing the easiness results of Elkind et al. [9].

## References

- [1] K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*. Elsevier, 2002.
- [2] S. Barberá and B. Moreno. Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *Games and Economic Behavior*. To appear.
- [3] J. Bartholdi, III and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4):165–169, 1986.
- [4] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.
- [5] K. Booth and G. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *JCSS*, 13(3):335–379, 1976.
- [6] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proc. of AAAI-10*, pages 715–722. AAAI Press, July 2010.
- [7] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009.
- [8] A. Ehrenfeucht, T. Harju, and G. Rozenberg. *The Theory of 2-Structures: A Framework for Decomposition and Transformation of Graphs*. World Scientific, 1999.
- [9] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proc. of AAAI-10*, pages 768–773. AAAI Press, July 2010.
- [10] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

- [11] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proc. of ECAI-08*, pages 366–370, July 2008.
- [12] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53(11):74–82, 2010.
- [13] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.
- [14] J. Gans and M. Smart. Majority voting with single-crossing preferences. *Journal of Public Economics*, 59:219–237, 1996.
- [15] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.
- [16] G. Laffond, J. Laine, and J. Laslier. Composition consistent tournament solutions and social choice functions. *Social Choice and Welfare*, 13(1):75–93, 1996.
- [17] J. Laslier. Rank-based choice correspondencies. *Economics Letters*, 52(3):279–286, 1996.
- [18] J. Laslier. Aggregation of preferences with a variable set of alternatives. *Social Choice and Welfare*, 17(2):269–282, 2000.
- [19] J. Mirrlees. An exploration in the theory of optimal income taxation. *Review of Economic Studies*, 38:175–208, 1971.
- [20] R. Möhring. Algorithmic aspects of the substitution decomposition in optimization over relations, set systems and boolean functions. *Annals of Op. Res.*, 4:195–225, 1985.
- [21] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008.
- [22] A. Saporiti and F. Tohmé. Single-crossing, strategic voting and the median choice rule. *Social Choice and Welfare*, 26(2):363–383, 2006.
- [23] M. Satterthwaite. Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.
- [24] T. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.
- [25] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proc. of AAAI-07*, pages 3–8, July 2007.

## A Material Missing from Section 3

Before we move on to the proofs, we need several additional definitions.

**Definition A.1.** Let  $C$  be a set of candidates.

1. Given three pairwise disjoint subsets  $X, Y, Z$  of  $C$ , and an order  $\succ$  over  $C$ , we say that  $X$  separates  $Y$  and  $Z$  in  $\succ$  if either  $Y \succ X \succ Z$  or  $Z \succ X \succ Y$ .
2. We say that an alternative  $a \in C$  splits a subset  $X \subseteq C$  with respect to an order  $\succ$  if  $X$  can be partitioned into two nonempty sets  $X_1$  and  $X_2$  such that  $\{a\}$  separates  $X_1$  and  $X_2$  in  $\succ$ ; note that this implies  $a \notin X$ .

**Proposition A.2.** Given a profile  $\mathcal{R} = (R_1, \dots, R_n)$ , let  $\mathcal{R}' = (R'_1, \dots, R'_n)$  be a profile such that  $R'_i \in \{R_i, \overleftarrow{R_i}\}$  for all  $i = 1, \dots, n$ . Then  $\mathcal{C}(\mathcal{R}) = \mathcal{C}(\mathcal{R}')$ .

**Proposition 3.3.** Let  $\mathcal{R}$  be a profile on  $[m]$ . Then (1)  $\{i\} \in \mathcal{C}(\mathcal{R})$  for any  $i \in [m]$ ; (2)  $\emptyset \notin \mathcal{C}(\mathcal{R})$  and  $[m] \in \mathcal{C}(\mathcal{R})$ ; (3) if  $C_1$  and  $C_2$  are in  $\mathcal{C}(\mathcal{R})$  and  $C_1 \cap C_2 \neq \emptyset$ , then  $C_1 \cup C_2$  and  $C_1 \cap C_2$  are also in  $\mathcal{C}(\mathcal{R})$ ; (4) if  $C_1$  and  $C_2$  are in  $\mathcal{C}(\mathcal{R})$  and  $C_1 \bowtie C_2$ , then  $C_1 \setminus C_2$  and  $C_2 \setminus C_1$  are also in  $\mathcal{C}(\mathcal{R})$ .

*Proof.* Properties (1) and (2) are immediate. Let us prove (3). Let  $C_1$  and  $C_2$  be two sets in  $\mathcal{C}(\mathcal{R})$  with  $I = C_1 \cap C_2 \neq \emptyset$ , and let  $R_i$  be an arbitrary preference order from  $\mathcal{R}$ . Consider some  $a \in [m]$ . Since  $C_1 \in \mathcal{C}(\mathcal{R})$ , if  $a \in [m] \setminus C_1$ , then  $a$  does not split  $C_1$  and, as a result,  $a$  does not split  $I$ . Similarly, no alternative in  $[m] \setminus C_2$  can split  $I$ . Thus, no element outside of  $I$  can split  $I$ , and hence members of  $I$  are ranked contiguously in  $R_i$ . Since this holds for any  $R_i$  in  $\mathcal{R}$ , we have  $I \in \mathcal{C}(\mathcal{R})$ .

Now, suppose that there is an alternative  $a \in [m] \setminus (C_1 \cup C_2)$  that splits  $C_1 \cup C_2$  in some order  $R_i$ . We know that  $a$  splits neither  $C_1$  nor  $C_2$ , hence either  $C_1 \succ_i a \succ_i C_2$  or  $C_2 \succ_i a \succ_i C_1$ , which is impossible since the intersection of  $C_1$  and  $C_2$  is nonempty. Thus,  $C_1 \cup C_2 \in \mathcal{C}(\mathcal{R})$ . This proves that (3) holds.

Let us now consider property (4). Suppose  $C_1, C_2 \in \mathcal{C}(\mathcal{R})$  and  $C_1 \bowtie C_2$ . Consider the set  $C_1 \setminus C_2$ ; for  $C_2 \setminus C_1$  the argument is similar. First, no element outside of  $C_1$  can split  $C_1 \setminus C_2$ , because otherwise it will split  $C_1$  too. Further, in each  $R_i$ , the intersection  $C_1 \cap C_2$  separates  $C_1 \setminus C_2$  and  $C_2 \setminus C_1$ . Hence elements of  $C_1 \cap C_2$  cannot split  $C_1 \setminus C_2$  either, and the property follows.  $\square$

**Proposition 3.4.** For any profile  $\mathcal{R}$  on  $[m]$ , each  $X \in \mathcal{C}(\mathcal{R})$  has at most two proper minimal supersets in  $\mathcal{C}(\mathcal{R})$ .

*Proof.* For the sake of contradiction, assume that there are three distinct sets  $Y, Z, W$  in  $\mathcal{C}(\mathcal{R})$  such that each of them is a proper minimal superset of  $X$ . It is easy to see that  $Y \cap Z = X$ : by Proposition 3.3,  $Y \cap Z \in \mathcal{C}(\mathcal{R})$ , so if  $(Y \cap Z) \setminus X \neq \emptyset$ , neither  $Y$  nor  $Z$  would be a proper minimal superset of  $X$ . Similarly,  $Y \cap W = X$  and  $Z \cap W = X$ . Pick two alternatives  $y, z$  so that  $y \in Y \setminus X$  and  $z \in Z \setminus X$ . Let  $R_i$  be a preference order from



$\mathcal{R}$ . The set  $X$  separates  $Y \setminus X$  and  $Z \setminus X$ , and so either  $y \succ_i X \succ_i z$  or  $z \succ_i X \succ_i y$ ; by Proposition A.2 we may assume the former. Now, pick  $w \in W \setminus X$ . A similar argument shows that we have  $y \succ_i X \succ_i w$  (as  $w \succ_i X \succ_i y$  leads to a contradiction). But now we must have  $z \succ_i X \succ_i w$  or  $w \succ_i X \succ_i z$ , none of which is possible.  $\square$

**Proposition 3.6.** *If  $\mathcal{C}$  is a clone structure, it does not contain a bicycle chain.*

*Proof.* Suppose that a clone structure  $\mathcal{C}$  contains a bicycle chain  $\{A_0, \dots, A_{k-1}\}$ , and let  $\mathcal{R} = (R_1, \dots, R_n)$  be a preference profile such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ .

As argued in the proof of Proposition 3.3, the set  $A_0 \cap A_1$  separates  $A_0 \setminus A_1$  and  $A_1 \setminus A_0$  in  $R_1$ . Thus, by Proposition A.2, we can assume that we have  $A_0 \setminus A_1 \succ_1 A_0 \cap A_1 \succ_1 A_1 \setminus A_0$ . Further, by the definition of the bicycle chain we have  $A_1 \setminus A_0 = A_1 \cap A_2$ ,  $A_1 \setminus A_2 = A_0 \cap A_1$ . Again, we have  $A_1 \cap A_0 = A_1 \setminus A_2 \succ_1 A_1 \cap A_2 \succ_1 A_2 \setminus A_1$ . Now, if  $k = 3$ , we have a contradiction already: since  $A_0 \setminus A_1 \neq \emptyset$  and  $A_0 \subseteq A_1 \cup A_2$ , it has to be the case that  $A_0$  intersects  $A_2$ , yet all elements of  $A_2$  are ranked strictly below  $A_0$ . If  $k > 3$ , continuing inductively, we obtain that for each  $i = 1, \dots, k-1$  the set  $A_i$  is ranked below  $A_{i-1} \setminus A_i$  in  $R_1$ . Hence, all elements of  $A_{k-1}$  are ranked below  $A_0$  in  $R_1$ . However, we have  $A_0 \cap A_{k-1} \neq \emptyset$ , a contradiction.  $\square$

**Proposition 3.8.** *Let  $\mathcal{E}$  and  $\mathcal{F}$  be families of subsets on disjoint sets  $E$  and  $F$ , respectively, that satisfy A1–A5. Then for any  $e \in E$  the set family  $\mathcal{E}(e \rightarrow \mathcal{F})$  also satisfies A1–A5.*

*Proof.* We have  $\{e'\} \in \mathcal{E}$  for all  $e' \in E$ ,  $\{f\} \in \mathcal{F}$  for all  $f \in F$ , so  $\{g\} \in \mathcal{E}(e \rightarrow \mathcal{F})$  for all  $g \in (E \setminus \{e\}) \cup F$ . Clearly,  $\emptyset \notin \mathcal{E}(e \rightarrow \mathcal{F})$ . Further,  $E \in \mathcal{E}$  and  $e \in E$ , so  $(E \setminus \{e\}) \cup F \in \mathcal{E}(e \rightarrow \mathcal{F})$ . Thus, A1 is satisfied.

Throughout the rest of the proof, we will use the observation that no set  $D \in \mathcal{E}(e \rightarrow \mathcal{F})$  can intersect  $F$  non-trivially, i.e., we have that for each  $D \in \mathcal{E}(e \rightarrow \mathcal{F})$  it holds that either  $D \cap F = \emptyset$  or  $D \setminus F = \emptyset$  or  $F \setminus D = \emptyset$ .

We will now show that  $\mathcal{E}(e \rightarrow \mathcal{F})$  satisfies A3 and A4. Consider two sets  $C_1, C_2$  in  $\mathcal{E}(e \rightarrow \mathcal{F})$  such that  $C_1 \cap C_2 \neq \emptyset$ . If  $C_1 \subseteq C_2$  or  $C_2 \subseteq C_1$ , then A2 and A3 trivially hold, so we can assume that  $C_1 \not\subseteq C_2$  and  $C_2 \not\subseteq C_1$ .

Suppose first that  $C_1 \subseteq F$ . Then  $C_2 \cap F \neq \emptyset$  and it cannot be the case that  $F \subseteq C_2$ , since we assume  $C_1 \setminus C_2 \neq \emptyset$ . Hence,  $C_2 \subseteq F$ , so  $C_1, C_2 \in \mathcal{F}$ , and the sets  $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$  belong to  $\mathcal{F}$  and hence to  $\mathcal{E}(e \rightarrow \mathcal{F})$ .

Next, suppose that  $F \subseteq C_1$ . Set  $C'_1 = (C_1 \setminus F) \cup \{e\}$ . Since  $C_2 \not\subseteq C_1$ , we have  $C_2 \not\subseteq F$ , and hence either  $F \subseteq C_2$  or  $F \cap C_2 = \emptyset$ . In the former case, set  $C'_2 = (C_2 \setminus F) \cup \{e\}$ ; in the latter case, set  $C'_2 = C_2$ . In both cases, we have  $C'_1, C'_2 \in \mathcal{E}$ , and  $C'_1 \not\subseteq C'_2$  and  $C'_2 \not\subseteq C'_1$ . Therefore, the sets  $C'_1 \cap C'_2, C'_1 \cup C'_2, C'_1 \setminus C'_2, C'_2 \setminus C'_1$  belong to  $\mathcal{E}$ , and hence the sets  $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$  belong to  $\mathcal{E}(e \rightarrow \mathcal{F})$ .

If  $F \subseteq C_2$ , the argument is similar. Thus, it remains to consider the case  $C_1 \cap F = \emptyset$ ,  $C_2 \cap F = \emptyset$ . Then  $C_1, C_2 \in \mathcal{E}$ . Thus, the sets  $C_1 \cap C_2, C_1 \cup C_2, C_1 \setminus C_2, C_2 \setminus C_1$  belong to  $\mathcal{E}$  and do not contain  $e$ , and hence they belong to  $\mathcal{E}(e \rightarrow \mathcal{F})$  as well. Thus, axioms A2 and A3 are satisfied.

To show that A4 holds, assume for the sake of contradiction that some set  $C \in \mathcal{E}(e \rightarrow \mathcal{F})$  has three proper minimal supersets  $X, Y$ , and  $Z$  in  $\mathcal{E}(e \rightarrow \mathcal{F})$ . If we have  $C \subseteq F$ , then the sets  $X, Y$  and  $Z$  cannot strictly contain  $F$  (or they would not be proper minimal supersets), but have to intersect  $F$ , so it has to be the case that  $X, Y, Z \subseteq F$ . Thus,  $C$  has three proper minimal supersets in  $\mathcal{F}$ , a contradiction. Next, suppose that  $F \subseteq C$ . Then all three sets  $X, Y$  and  $Z$  are supersets of  $F$ , too. Consider the sets  $C' = (C \setminus F) \cup \{e\}$ ,  $X' = (X \setminus F) \cup \{e\}$ ,  $Y' = (Y \setminus F) \cup \{e\}$ ,  $Z' = (Z \setminus F) \cup \{e\}$ . All these sets are in  $\mathcal{E}$ . Moreover,  $X', Y'$  and  $Z'$  are distinct, and each of them is a superset of  $C'$ . To see that all of them are proper supersets of  $C'$ , observe that if  $C' \subset T' \subset X'$ , then  $C \subset (T' \setminus \{e\}) \cup F \subset X$ , a contradiction with  $X$  being a minimal proper superset of  $C$ . Thus,  $C'$  has three minimal proper supersets in  $\mathcal{E}$ , a contradiction. Finally, if  $C \cap F = \emptyset$ , we have  $C \in \mathcal{E}$ . For  $T = X, Y, Z$ , let  $T' = T$  if  $F \cap T = \emptyset$  and  $T' = (T \setminus F) \cup \{e\}$  otherwise. Clearly, the sets  $X', Y'$  and  $Z'$  are in  $\mathcal{E}$ . By the same argument as above, we can show that  $C$  has three minimal proper supersets in  $\mathcal{E}$ , a contradiction. Thus,  $\mathcal{E}(e \rightarrow \mathcal{F})$  satisfies A4.

Finally, let  $\mathcal{E}(e \rightarrow \mathcal{F})$  contain a bicycle chain  $\{A_0, \dots, A_{k-1}\}$ ; in what follows, all indices are computed modulo  $k$ . Suppose first that we have  $A_i \subseteq F$  for some  $i = 0, \dots, k-1$ . Then  $A_{i-1} \cap F \neq \emptyset$ ,  $A_{i+1} \cap F \neq \emptyset$ . Since both  $A_{i-1}$  and  $A_{i+1}$  intersect  $A_i$  non-trivially, neither of them can contain  $F$ , and therefore both of them are subsets of  $F$ . Applying this argument inductively, we conclude that all sets  $A_i$ ,  $i = 0, \dots, k-1$ , are subsets of  $F$ , i.e.,  $\mathcal{F}$  contains a bicycle chain, a contradiction. Thus, we can assume that for each  $i = 0, \dots, k-1$  either  $F \subseteq A_i$  or  $F \cap A_i = \emptyset$ . For each  $i = 0, \dots, k-1$ , set  $A'_i = (A_i \setminus F) \cup \{e\}$  if  $F \subseteq A_i$  and set  $A'_i = A_i$  otherwise. It is straightforward to check that the set family  $\{A'_0, \dots, A'_{k-1}\}$  is a bicycle chain in  $\mathcal{E}$ , a contradiction. Thus,  $\mathcal{E}(e \rightarrow \mathcal{F})$  satisfies A5.  $\square$

**Proposition A.3.** *Let  $\mathcal{F}$  be an irreducible family of subsets of  $[m]$  that satisfies A1–A5, and let  $D$  be a minimal proper subset of  $\mathcal{F}$ . Then  $|D| = 2$ .*

*Proof.* Suppose for the sake of contradiction that  $|D| \geq 3$ . The set family  $\mathcal{D} = \{F \in \mathcal{F} \mid F \subseteq D\}$  is not a subfamily of  $\mathcal{F}$ , which means that  $\mathcal{F}$  contains a proper subset  $E$  such that  $D \bowtie E$ . However, by A2 and A3, both  $D \cap E$  and  $D \setminus E$  must belong to  $\mathcal{F}$ , both are strict subsets of  $D$ , and at least one of them has at least two elements. Thus,  $D$  is not a minimal proper subset, a contradiction.  $\square$

**Proposition A.4.** *Let  $\mathcal{F}$  be a family of subsets of  $[m]$  that satisfies A1–A5. Then each candidate  $i \in [m]$  belongs to at most two minimal proper subsets in  $\mathcal{F}$ .*

*Proof.* Suppose for the sake of contradiction, that  $i$  belongs to three minimal proper subsets in  $\mathcal{F}$ . Since these subsets are minimal proper subsets, they are also minimal proper supersets of  $\{i\}$ . However, by A4, no subset of  $\mathcal{F}$  has more than two minimal proper supersets, a contradiction.  $\square$

**Theorem 3.10.** *Any irreducible family of subsets satisfying A1–A5 is either a string of sausages or a fat sausage.*

*Proof.* Let  $\mathcal{F}$  be an irreducible family of subsets over  $[m]$  that satisfies A1–A5. If  $\mathcal{F}$  does not contain any proper subsets, then it is a fat sausage. Thus, for the remainder of the proof let us assume that  $\mathcal{F}$  does contain at least one proper subset.

Let us consider a graph  $G$  whose vertices are elements of  $[m]$  and there is an edge between  $i$  and  $j$  if and only if  $\{i, j\}$  is a minimal proper subset of  $\mathcal{F}$ . By Proposition A.4, the degree of each vertex in  $G$  is at most 2. Further,  $G$  cannot contain cycles, since each cycle in  $G$  would correspond to a bicycle chain in  $\mathcal{F}$  formed by the two-element subsets  $\{i, j\}$ . Thus,  $G$  is a collection of paths. We will now prove that  $G$  has at most one connected component, and hence  $\mathcal{F}$  is a string of sausages.

Let  $G'$  be a maximal connected component in  $G$ , and let  $F$  be the set of vertices of  $G'$ . Suppose that  $F \neq [m]$ . Note that by A3,  $F$  is a subset in  $\mathcal{F}$ . Since  $\mathcal{F}$  is not a fat sausage, by Proposition A.3 we have  $|F| \geq 2$ . Let us rename the alternatives so that  $F = \{f_1, \dots, f_k\}$  and each  $\{f_i, f_{i+1}\}$ ,  $1 \leq i < k$ , is an edge of  $G'$ .

If  $F \neq [m]$ , there exists a proper subset  $E \in \mathcal{F}$  such that  $E \bowtie F$ . Let us pick such a set  $E$  for which  $|E \setminus F|$  is smallest. By A3, the set  $E \setminus F$  belongs to  $\mathcal{F}$ . We consider two cases.

**$|\mathbf{E} \setminus \mathbf{F}| = 1$ .** Observe that in this case  $|E \cap F| \geq 2$ : otherwise,  $E$  would be an edge of  $G'$ . Let  $e$  be a member of  $E \setminus F$ . Suppose first that  $E \cap F$  is not a contiguous subset of  $F$ , that is, there are some  $i, j, \ell \in [k]$  such that  $i < \ell < j$ , and (i)  $f_i \in E$  and  $f_s \notin E$  for  $s < i$ , (ii)  $f_j \in E$  and  $f_t \notin E$  for  $t > j$ , and (iii)  $f_\ell \notin E$ . Then either  $E \cap F = \{f_i, f_j\}$ , or  $E \cap F$  intersects  $\{f_{i+1}, \dots, f_{j-1}\}$  and we have  $\{f_i, f_j\} = (E \cap F) \setminus \{f_{i+1}, \dots, f_{j-1}\}$ . In both cases, we can use axiom A3 to conclude that  $\{f_i, f_j\}$  belongs to  $\mathcal{F}$ , and hence  $G'$  contains a cycle, a contradiction. Thus, we have  $E \cap F = \{f_i, \dots, f_j\}$  for some  $1 \leq i < j \leq k$ .

Suppose that  $j \neq k$ . Then, since  $i < j$ , by A3 the set  $E \setminus \{f_1, \dots, f_{j-1}\} = \{e, f_j\}$  is in  $\mathcal{F}$ . However, this means that  $e \in F$ , which is a contradiction. Thus,  $j = k$ . Similarly, we can argue that  $i = 1$ . Hence, we have  $F \subseteq E$ , a contradiction.

**$|\mathbf{E} \setminus \mathbf{F}| > 1$ .** By A3,  $E \setminus F$  is a proper subset in  $\mathcal{F}$ . Thus, since  $\mathcal{F}$  is irreducible, there is a proper subset  $H$  in  $\mathcal{F}$  such that  $H \bowtie (E \setminus F)$ .

Suppose first that  $F \subseteq H$ , and consider the set  $H' = H \cap E$ . Since  $E$  intersects  $F$ , we have  $H' \cap F \neq \emptyset$ . Further,  $H' \cap (E \setminus F) = H \cap (E \setminus F) \neq \emptyset$ , so  $H' \setminus F \neq \emptyset$ . Finally,  $F \setminus E \neq \emptyset$  and  $H' \subseteq E$ , so  $F \setminus H' \neq \emptyset$ . Thus,  $F \bowtie H'$ . However,  $H' \setminus F = H \cap (E \setminus F)$  is a strict subset of  $E \setminus F$ , so  $|H' \setminus F| < |E \setminus F|$ , a contradiction with our choice of  $E$ .

Thus, we have  $F \not\subseteq H$ . If, nevertheless,  $F \cap H \neq \emptyset$ , we set  $H'' = H \cap (E \cup F)$ . Clearly, we have  $F \cap H'' \neq \emptyset$ . Since  $H''$  is a subset of  $H$ , we also have  $F \setminus H'' \neq \emptyset$ . Finally, since  $H \cap (E \setminus F) \neq \emptyset$ , we have  $H'' \setminus F \neq \emptyset$ . Thus,  $H'' \bowtie F$ , yet  $H'' \setminus F = H \cap (E \setminus F)$  is a strict subset of  $E \setminus F$ , so  $|H'' \setminus F| < |E \setminus F|$ , a contradiction with our choice of  $E$ .

Hence,  $H \cap F = \emptyset$ . However, this means that  $E \setminus H$  still intersects  $F$  nontrivially, and  $|(E \setminus H) \setminus F| < |E \setminus F|$ , a contradiction again.

We have shown that assuming that  $F \neq [m]$  leads to a contradiction. Hence,  $F = [m]$ , which means that  $\mathcal{F}$  is a string of sausages.  $\square$

**Proposition 3.11.** *Let  $\mathcal{C}$  and  $\mathcal{D}$  be two clone structures over sets  $C$  and  $D$ , respectively, where  $|C| = m$ ,  $|D| = k$ , and  $C \cap D = \emptyset$ . Then for each  $c \in C$ , the family of subsets*

$\mathcal{C}(c \rightarrow \mathcal{D})$  is a clone structure.

*Proof.* Fix a candidate  $c \in C$ , and let  $\mathcal{R} = (R_1, \dots, R_n)$  and  $\mathcal{Q} = (Q_1, \dots, Q_{n'})$  be two profiles of voters such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$  and  $\mathcal{D} = \mathcal{C}(\mathcal{Q})$ . Since duplicating linear orders in  $\mathcal{R}$  and  $\mathcal{Q}$  does not change  $\mathcal{C}$  and  $\mathcal{D}$ , we can assume without loss of generality that  $n = n' \geq 2$ . Our goal is to construct a profile  $\mathcal{R}'$  such that  $\mathcal{C}(c \rightarrow \mathcal{D}) = \mathcal{C}(\mathcal{R}')$ . This profile will have  $n$  voters and  $m + k - 1$  alternatives, that is,  $\mathcal{R}' = (R'_1, \dots, R'_n)$ . We will construct  $\mathcal{R}'$  in two steps. First, for each  $i = 1, \dots, n$ , we set  $R'_i$  to be identical to  $R_i$  except that the occurrence of  $c$  is replaced by  $Q_i$ ; denote the resulting profile by  $\mathcal{R}^0 = (R_1^0, \dots, R_n^0)$  and let  $\mathcal{C}^0 = \mathcal{C}(\mathcal{R}^0)$ . It is easy to see that all elements of  $\mathcal{C}(c \rightarrow \mathcal{D})$  are clones in  $\mathcal{R}^0$ , so  $\mathcal{C}(c \rightarrow \mathcal{D}) \subseteq \mathcal{C}^0$ . If also  $\mathcal{C}^0 \subseteq \mathcal{C}(c \rightarrow \mathcal{D})$ , we are done, since in this case we can set  $\mathcal{R}' = \mathcal{R}^0$ .

Otherwise, we flip  $Q_n$ . That is, assuming without loss of generality that  $Q_n$  ranks the elements of  $D$  as

$$Q_n : d_1 \succ d_2 \succ \dots \succ d_k$$

and  $R_n$  is given by  $C_1 \succ c \succ C_2$ , we define

$$R'_n : C_1 \succ d_k \succ \dots \succ d_1 \succ C_2,$$

where we assume that  $R'_n$  orders the elements of  $C_1$  and  $C_2$  in the same way as  $R_n$  does; we also set  $R'_i = R_i^0$  for  $i = 1, \dots, n-1$ . Consider the resulting profile  $\mathcal{R}'$ , and let  $\mathcal{C}' = \mathcal{C}(\mathcal{R}')$ . We claim that  $\mathcal{C}' = \mathcal{C}(c \rightarrow \mathcal{D})$ . As above, it is easy to see that  $\mathcal{C}(c \rightarrow \mathcal{D}) \subseteq \mathcal{C}'$ . It remains to show that  $\mathcal{C}' \subseteq \mathcal{C}(c \rightarrow \mathcal{D})$ .

Let  $X$  be a “parasite” clone in  $\mathcal{C}^0 \setminus \mathcal{C}(c \rightarrow \mathcal{D})$ . Clearly, it cannot be the case that  $X \subseteq C$  or  $X \subseteq D$ . Further, if  $D \subseteq X$ , then  $(X \setminus D) \cup \{c\}$  is a clone in  $\mathcal{C}$ , and hence  $X \in \mathcal{C}(c \rightarrow \mathcal{D})$ . Thus, the sets  $C_X = X \cap C$  and  $D_X = X \cap D$  are both non-empty, and  $D_X \neq D$ . By Proposition A.2, we may assume that each order in  $\mathcal{R}^0$  is of the form  $\dots \succ C_X \succ D_X \succ D \setminus D_X \succ \dots$ .

Now, suppose for the sake of contradiction that  $Y$  is a clone in  $\mathcal{C}' \setminus \mathcal{C}(c \rightarrow \mathcal{D})$ . By the same argument as in the previous paragraph, we conclude that  $Y \cap D \neq \emptyset$ ,  $Y \cap C \neq \emptyset$ , and  $D \not\subseteq Y$ . Thus, we have two possibilities:

- $d_1 \in Y$ ,  $d_k \notin Y$ . Then, since  $Y$  is contiguous in  $R'_1$  and  $Y \cap C \neq \emptyset$ , we have  $Y \cap C_X \neq \emptyset$ . However, in  $R'_n$  the element  $d_k$  splits  $Y$  and  $C_X$ , a contradiction.
- $d_1 \notin Y$ ,  $d_k \in Y$ . Then, since  $Y$  is contiguous in  $R'_n$  and  $Y \cap C \neq \emptyset$ , we have  $Y \cap C_X \neq \emptyset$ . However, in  $R'_1$  the element  $d_1$  splits  $Y$  and  $C_X$ , a contradiction.

Hence, we have  $Y \in \mathcal{C}(c \rightarrow \mathcal{D})$ . The proof is complete.  $\square$

The above proof could be simplified if we were willing to use more voters in the profile for  $\mathcal{C}(c \rightarrow \mathcal{D})$ . However, the current version of the proof is very useful when we consider the number of voters needed to implement a particular clone structure.

**Theorem 3.12.** *A family  $\mathcal{F}$  of subsets of  $[m]$  is a clone structure if and only if it satisfies conditions A1–A5.*

*Proof.* We have already argued that any clone structure satisfies A1–A5; it remains to prove that the converse is also true.

Our proof is by induction on  $m$ . Clearly the theorem holds for  $m = 1$  and for  $m = 2$ . For the inductive step, assume it holds for each  $m' < m$ . Let  $\mathcal{F}$  be a family of subsets of  $[m]$  that satisfies A1–A5. If  $\mathcal{F}$  is irreducible then, by Theorem 3.10, it is either a string of sausages or a fat sausage and thus a clone structure. Otherwise,  $\mathcal{F}$  contains a proper subfamily  $\mathcal{D}$ . Let  $\mathcal{F}' = \mathcal{F}(\mathcal{D} \rightarrow e)$  for some  $e \notin [m]$ . We have argued that  $\mathcal{D}$  and  $\mathcal{F}'$  satisfy axioms A1–A5. Hence, by our inductive hypothesis both  $\mathcal{F}'$  and  $\mathcal{D}$  are clone structures and so, by Proposition 3.11,  $\mathcal{F} = \mathcal{F}'(e \rightarrow \mathcal{D})$  is a clone structure as well. This completes the proof.  $\square$

## A.1 Identifying Clone Structures in P

**Theorem A.5.** *There exists a polynomial-time algorithm that, given a family of subsets  $\mathcal{F}$  over a finite set  $F$ , checks if  $\mathcal{F}$  is a clone structure over  $F$ .*

*Proof.* It is easy to see that we can check in polynomial time whether  $\mathcal{F}$  satisfies A1–A4. Now, suppose that  $\mathcal{F}$  has passed this check, and it remains to verify that it satisfies A5. We can directly check if  $\mathcal{F}$  contains a bicycle chain of size 3, by considering all possible triples of the subsets in  $\mathcal{F}$ . To check for bicycle chains of size 4 or more, we will construct a directed graph  $G$  as follows.

The vertices of  $G$  are ordered pairs  $(X, Y)$ , where  $X$  and  $Y$  are two subsets in  $\mathcal{F}$  such that  $X \bowtie Y$ . There is a directed edge from  $(X, Y)$  to  $(Y', Z)$  if  $Y = Y'$ ,  $X \cap Y \cap Z = \emptyset$ , and  $Y \subseteq X \cup Z$ . Intuitively,  $G$  has an edge from  $(X, Y)$  to  $(Y, Z)$  if  $X$ ,  $Y$  and  $Z$  can be three consecutive sets in a bicycle chain. It is not hard to verify that  $G$  contains a directed cycle if and only if  $\mathcal{F}$  contains a bicycle chain of size 4 or more. Indeed, let  $\{A_0, \dots, A_{k-1}\}$  be a bicycle chain of size  $k \geq 4$  in  $\mathcal{F}$ . Then any pair  $(A_i, A_{i+1})$  is a vertex of  $G$ . Moreover, there is an edge in  $G$  between  $(A_{i-1}, A_i)$  and  $(A_i, A_{i+1})$ , so  $(A_0, A_1), (A_1, A_2), \dots, (A_{k-1}, A_0)$  is a directed cycle in  $G$  (as always in our discussion of bicycle chains, the indices are computed modulo  $k$ ). Conversely, if  $G$  contains a directed cycle of the form  $(X_0, X_1), (X_1, X_2), \dots, (X_{k-1}, X_0)$ , then the sets  $X_0, \dots, X_{k-1}$  form a bicycle chain.  $\square$

We can strengthen the proof of Theorem A.5 to obtain a logarithmic space algorithm. Verifying axioms A1–A4 in logarithmic space is straightforward. For axiom A5 the verification problem can be reduced to connectivity testing for undirected graphs; it then remains to apply the breakthrough result of Reingold [21].

## B Material Missing from Section 4

We will now prove Theorem 4.1. However, to do so, we need the following proposition.

**Proposition B.1.** *Let  $\mathcal{C}$  be an irreducible clone structure over  $[m]$ . If  $\mathcal{C}$  is a string of sausages, it is 1-implementable. If  $\mathcal{C}$  is a fat sausage and  $m > 3$ , then  $\mathcal{C}$  is 2-implementable,*

but not 1-implementable. If  $\mathcal{C}$  is a fat sausage and  $m = 3$ , then  $\mathcal{C}$  is 3-implementable, but not 2-implementable.

*Proof.* If  $\mathcal{C}$  is a string of sausages, it can be implemented using a single order, namely,  $1 \succ \dots \succ m$ .

Now, suppose that  $\mathcal{C}$  is a fat sausage. Clearly, it cannot be implemented with a single order, as the clone structure that corresponds to the latter is a string of sausages.

Suppose first that  $m = 2k$ . For convenience, set  $x_i = i$ ,  $y_i = k + i$  for  $i = 1, \dots, k$ . We define  $\mathcal{R} = (R_1, R_2)$  as follows.

$$\begin{aligned} R_1 : x_1 \succ \dots \succ x_k \succ y_1 \succ \dots \succ y_k, \\ R_2 : y_1 \succ x_1 \succ y_2 \succ x_2 \succ \dots \succ y_k \succ x_k. \end{aligned}$$

We claim that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ . Clearly, we have  $\mathcal{C} \subseteq \mathcal{C}(\mathcal{R})$ . Now, suppose that  $D \in \mathcal{C}(\mathcal{R}) \setminus \mathcal{C}$ , i.e.,  $|D| \neq 1, m$ . Since  $D$  has to be contiguous in  $R_1$ , we have one of the following three cases:

- (a)  $D = \{x_i, \dots, x_j\}$  for some  $1 \leq i < j \leq k$ ;
- (b)  $D = \{y_i, \dots, y_j\}$  for some  $1 \leq i < j \leq k$ ;
- (c)  $D = \{x_i, \dots, y_j\}$  for some  $1 \leq i \leq k, 1 \leq j \leq k$ .

Case (a) is impossible since in  $R_2$  the element  $y_j$  appears between  $x_i$  and  $x_j$ . Similarly, case (b) is impossible since in  $R_2$  the element  $x_i$  appears between  $y_i$  and  $y_j$ . In case (c) we have  $x_k, y_1 \in D$ . Since these elements appear at the opposite ends of  $R_2$ , we conclude that  $D = [m]$ , a contradiction.

Next, suppose that  $m = 2k + 1$ ,  $k > 1$ . Set  $x_i = i$ ,  $y_i = k + i$  for  $i = 1, \dots, k$ ,  $z = 2k + 1$ . We define  $\mathcal{R} = (R_1, R_2)$  as follows.

$$\begin{aligned} R_1 : x_1 \succ \dots \succ x_k \succ y_1 \succ \dots \succ y_{k-1} \succ z \succ y_k, \\ R_2 : y_1 \succ x_1 \succ y_2 \succ x_2 \succ \dots \succ y_k \succ x_k \succ z. \end{aligned}$$

Again, it is clear that  $\mathcal{C} \subseteq \mathcal{C}(\mathcal{R})$ . Now, suppose that  $D \in \mathcal{C}(\mathcal{R}) \setminus \mathcal{C}$ , i.e.,  $|D| \neq 1, m$ . As in the case of even  $m$ ,  $D$  cannot be of the form  $\{x_i, \dots, x_j\}$  for  $1 \leq i < j \leq k$ , or of the form  $\{y_i, \dots, y_j\}$  for  $1 \leq i < j \leq k - 1$ . Further, if  $D$  is of the form  $\{x_i, \dots, y_j\}$  for some  $i = 1, \dots, k$  and some  $j = 1, \dots, k - 1$ , then  $D$  must contain all elements that appear between  $x_k$  and  $y_1$  in  $R_2$ , i.e., either  $D = [m]$  or  $D = [m] \setminus \{z\}$ , which is impossible. Now, if  $D$  contains  $z$ , it must also contain the only element that is adjacent to it in  $R_2$ , i.e.,  $x_k$ . As  $y_1$  appears between  $x_k$  and  $z$  in  $R_1$ , we have  $y_1 \in D$ . But then  $D = [m]$ , since  $y_1$  and  $z$  are extreme elements of  $R_2$ .

Finally, if  $m = 3$ , we can set  $\mathcal{R} = (R_1, R_2, R_3)$ , where  $R_1 : 1 \succ 2 \succ 3$ ,  $R_2 : 2 \succ 1 \succ 3$ ,  $R_3 : 2 \succ 3 \succ 1$ . To see that  $\mathcal{C}$  cannot be implemented by any 2-voter profile  $(R_1, R_2)$ , observe that we can assume without loss of generality that  $R_1$  is of the form  $1 \succ 2 \succ 3$ , and in  $R_2$  element 2 is adjacent to at least one of the remaining elements (and hence forms a clone with that element).  $\square$

Now we are ready to prove Theorem 4.1. However, we will first prove a stronger result, and only derive Theorem 4.1 as its corollary.

**Theorem B.2.** *Any clone structure  $\mathcal{C}$  is 3-implementable. Moreover, if the tree  $T(\mathcal{C})$  does not have nodes that carry labels of the form  $x \odot y \odot z$ , then  $\mathcal{C}$  is 2-implementable. If  $\mathcal{C}$  is a string of sausages then it is 1-implementable.*

*Proof.* If  $\mathcal{C}$  is a string of sausages, then it clearly is 1-implementable. Otherwise, the following argument proves the theorem.

Fix a clone structure  $\mathcal{C}$  on a set  $C$  of size  $m$ . If  $T(\mathcal{C})$  does not have nodes that carry labels of the form  $x \odot y \odot z$ , then set  $k = 2$ . Otherwise set  $k = 3$ . The proof is by induction on  $m$ . If  $m = 1$  or  $m = 2$ , the theorem is obviously true. Further, if  $\mathcal{C}$  is irreducible, the theorem follows from Proposition B.1. Otherwise,  $\mathcal{C}$  contains a proper subfamily  $\mathcal{D}$ . By the inductive assumption, the clone structures  $\mathcal{D}$  and  $\mathcal{C}(\mathcal{D} \rightarrow d)$ , where  $d \notin C$ , are  $k$ -implementable. Let  $\mathcal{R} = (R_1, \dots, R_k)$  and  $\mathcal{Q} = (Q_1, \dots, Q_k)$  be the respective preference profiles, i.e.,  $\mathcal{C}(\mathcal{D} \rightarrow d) = \mathcal{C}(\mathcal{R})$ ,  $\mathcal{D} = \mathcal{C}(\mathcal{Q})$ . Then the proof of Proposition 3.11 shows how to combine  $\mathcal{R}$  and  $\mathcal{Q}$  to obtain a preference profile  $\mathcal{R}'$  with  $k$  voters such that  $\mathcal{C} = \mathcal{C}(\mathcal{R}')$ .  $\square$

Now the proof of Theorem 4.1 is immediate.

**Theorem 4.1.** *Any clone structure is 3-implementable.*

*Proof.* Follows directly from Theorem B.2.  $\square$

## C Material Missing from Section 5

This part of the appendix contains the missing proofs and discussion regarding clone structures in single-peaked elections.

Given a profile  $\mathcal{R}$  with  $|\text{peak}(\mathcal{R})| \geq 2$  that is single-peaked with respect to  $>$ , we say that  $p_1$  and  $p_2$  are the *extreme peaks* of  $\mathcal{R}$  with respect to  $>$  if either  $p_1 > p > p_2$  for each  $p \in \text{peak}(\mathcal{R}) \setminus \{p_1, p_2\}$  or  $p_2 > p > p_1$  for each  $p \in \text{peak}(\mathcal{R}) \setminus \{p_1, p_2\}$ . We say that two candidates  $a, b \in C$  are *on the same side* of  $c \in C \setminus \{a, b\}$  in  $>$  if either  $(a > c \wedge b > c)$  or  $(c > a \wedge c > b)$ . Otherwise, we say that  $a$  and  $b$  are *on the opposite sides* of  $c$  in  $>$ . Given two orders  $>$  and  $>'$  over  $C$ , we say that  $>$  and  $>'$  *agree on*  $D \subseteq C$  if for each  $a, b \in D$  it holds that  $a > b$  if and only if  $a >' b$ .

The single-peakedness of a given preference profile can be witnessed by many different orders; for instance, if  $\mathcal{R}$  is single-peaked with respect to  $>$ , it is also single-peaked with respect to  $\check{>}$ . However, it turns out that these orders have the same extreme peaks and agree (up to an inversion) on all candidates between these peaks.

**Theorem C.1.** *Consider a preference profile  $\mathcal{R}$  over  $C$  with  $|\text{peak}(\mathcal{R})| \geq 2$  that is single-peaked with respect to two orders  $>$  and  $>'$ . Let  $p_1$  and  $p_2$  be the extreme peaks of  $\mathcal{R}$  with respect to  $>$  such that  $p_1 > p_2$ . Then  $p_1$  and  $p_2$  are also the extreme peaks of  $\mathcal{R}$  with respect to  $>'$ . Moreover, if  $p_1 >' p_2$ , then  $>$  and  $>'$  agree on the set  $P = \{c \mid p_1 > c > p_2\} \cup \{p_1, p_2\}$ .*

*Proof.* Fix two orders  $>$  and  $>'$  that both witness the single-peakedness of  $\mathcal{R}$ . Consider two candidates  $p, q \in \text{peak}(\mathcal{R})$ , and another candidate  $c \in C$ . We claim that either  $p$  and  $q$  are on the same side of  $c$  in both  $>$  and  $>'$ , or they are on the opposite sides of  $c$  in both  $>$  and  $>'$ . Indeed, suppose that this is not the case. Without loss of generality we assume that  $p > q > c$  and  $p >' c >' q$ . Now, consider a preference order  $R_i$  such that  $\text{peak}(R_i) = p$ . Since  $\mathcal{R}$  is single-peaked with respect to  $>$ , it must be the case that  $p \succ_i q \succ_i c$ ; on the other hand, since  $\mathcal{R}$  is single-peaked with respect to  $>'$ , we have  $p \succ_i c \succ_i q$ , a contradiction. Hence, either  $p$  and  $q$  are on the same side of  $c$  in both  $>$  and  $>'$ , or  $p$  and  $q$  are on the opposite sides of  $c$  in both  $>$  and  $>'$ .

Now, consider an arbitrary  $p \in \text{peak}(\mathcal{R}) \setminus \{p_1, p_2\}$ . Since  $p_1$  and  $p_2$  are the extreme peaks of  $\mathcal{R}$  and  $p_1 > p_2$ , we have  $p_1 > p > p_2$ . Therefore, by the argument above we have  $p_1 >' p >' p_2$ . This proves the first statement of the theorem.

To prove the second statement, assume that  $p_1 >' p_2$ . Also, without loss of generality, assume that  $p_1 = \text{peak}(R_1)$  and  $p_2 = \text{peak}(R_2)$ . Let  $P' = \{c \mid p_1 >' c >' p_2\} \cup \{p_1, p_2\}$ . Suppose that  $P \setminus P' \neq \emptyset$ , and consider a candidate  $c \in P \setminus P'$ . The candidates  $p_1$  and  $p_2$  are on the opposite sides of  $c$  in  $>$ , but on the same side of  $c$  in  $>'$ , a contradiction. Assuming  $P' \setminus P \neq \emptyset$  leads to a contradiction as well. Thus,  $P = P'$ . Now, suppose that for some  $c, d \in P \setminus \{p_1, p_2\}$  we have  $c > d$  and  $d >' c$ . Then, since  $\mathcal{R}$  is single-peaked with respect to  $>$ , we have  $p_1 \succ_1 c \succ_1 d$ . However, since  $\mathcal{R}$  is single-peaked with respect to  $>'$ , we have  $p_1 \succ_1 d \succ_1 c$ , a contradiction. Thus, the theorem is proved.  $\square$

Thus, by Theorem C.1, we can speak of extreme peaks of a single-peaked profile, without referring to a specific societal axis.

**Proposition 5.2.** *Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a preference profile over a candidate set  $C$  that is single-peaked with respect to some order  $>$ , and let  $D \in \mathcal{C}(\mathcal{R})$  be a clone set for  $\mathcal{R}$  with  $|D| \geq 2$ . Then  $C$  can be partitioned into pairwise disjoint sets  $A_1, A_2, D_1, D_2$ , and  $P$  so that  $C \setminus D = A_1 \cup P \cup A_2$ ,  $D = D_1 \cup D_2$ ,  $D_1 \neq \emptyset$ ,  $D_2 \neq \emptyset$ , and*

$$A_1 > D_1 > P > D_2 > A_2.$$

*Further, if  $P \neq \emptyset$ , then  $\text{peak}(\mathcal{R}) \subseteq P$ , and, moreover,  $P \succ_i D \succ_i A_1 \cup A_2$  for each  $i = 1, \dots, n$ .*

*Proof.* Suppose that our first claim is not true. Then there exist two candidates  $c_1, c_2 \in C \setminus D$  and three candidates  $d_1, d_2, d_3 \in D$  such that  $d_1 > c_1 > d_2 > c_2 > d_3$ . Let  $p = \text{peak}(R_1)$ . If  $p > d_2$  or  $p = d_2$ , we have  $d_2 \succ_1 c_2 \succ_1 d_3$  and hence  $D$  is not contiguous in  $R_1$ , a contradiction. Similarly, if  $d_2 > p$ , we have  $d_2 \succ_1 c_1 \succ_1 d_1$ , a contradiction again. This proves our claim regarding the partition of  $C$ . To see that we can ensure that both  $D_1$  and  $D_2$  are non-empty, note that if, e.g.,  $D_1 = \emptyset$ , we can modify the partition by merging  $P$  into  $A_1$  (so that the new  $P$  is empty), and repartitioning  $D$  into two non-empty sets (recall that  $|D| \geq 2$ ).

To prove the second claim, consider an arbitrary preference profile  $R_i$ . Since  $P \neq \emptyset$ , the peak of  $R_i$  must be in  $P$ , as otherwise it would be impossible for  $R_i$  to rank members



of  $D$  contiguously. Now, let us show that  $P \succ_i D$ . Suppose that this is not the case, i.e.,  $d \succ_i p$  for some  $d \in D$ ,  $p \in P$ . Since both  $D_1$  and  $D_2$  are non-empty, we can pick two alternatives  $d_1 \in D_1, d_2 \in D_2$ . As  $R_i$  ranks members of  $D$  contiguously, it has to be the case that  $D \succ_i p$ , and, in particular,  $\{d_1, d_2\} \succ_i p$ . But we have  $d_1 > p > d_2$ , so  $d_1 \succ_i p$  implies  $p \succ_i d_2$ , a contradiction. Thus  $P \succ_i D$ .

Finally, let us show that  $D \succ_i A_1 \cup A_2$ . Since the peak of  $R_i$  is in  $P$ ,  $A_1 > D_1$  implies  $D_1 \succ_i A_1$ , and  $D_2 > A_2$  implies  $D_2 \succ_i A_2$ . Suppose for the sake of contradiction that  $a \succ_i d$  for some  $a \in A$ ,  $d \in D$ . Then either  $a \in A_1$ ,  $d \in D_2$  (and hence  $D_1 \succ_i a \succ_i d$ ) or  $a \in A_2$ ,  $d \in D_1$  (and hence  $D_2 \succ_i a \succ_i d$ ). In both cases, we obtain a contradiction with  $D$  being contiguous in  $R_i$ . Thus,  $D \succ_i A_1 \cup A_2$ . This completes the proof.  $\square$

**Proposition C.2.** *Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a single-peaked preference profile over a candidate set  $C$ , and let  $D \in \mathcal{C}(\mathcal{R})$  be a clone set such that  $|D| \geq 2$ . Let  $c$  be some candidate not in  $C$ . Then the preference profile  $\mathcal{R}' = \mathcal{R}(D \mapsto c)$  is single-peaked.*

*Proof.* Suppose that  $\mathcal{R}$  is single-peaked with respect to some order  $>$  over  $C$ . We will now construct an order  $>'$  over  $(C \setminus D) \cup \{c\}$  such that  $\mathcal{R}'$  is single-peaked with respect to  $>'$ .

By Proposition 5.2, we know that  $C$  can be partitioned into  $P, D_1, D_2, A_1, A_2$  so that  $D_1 \neq \emptyset, D_2 \neq \emptyset$  and

$$A_1 > D_1 > P > D_2 > A_2.$$

If  $P = \emptyset$ , we set  $>'$  to be an order that agrees with  $>$  on  $A_1 \cup A_2$  and satisfies  $A_1 >' c >' A_2$ . If  $P \neq \emptyset$ , we set  $>'$  to be an order that agrees with  $>$  on  $A_1 \cup P \cup A_2$  and satisfies  $A_1 >' c >' P >' A_2$ . In both cases, it is immediate that  $\mathcal{R}'$  is single-peaked with respect to  $>'$ .  $\square$

**Proposition 5.7.** *Let  $\mathcal{R}$  be a preference profile over a set of candidates  $C$ . Let  $D^1, \dots, D^k \in \mathcal{C}(\mathcal{R})$  be a sequence of pairwise disjoint clone sets, and let  $c^1, \dots, c^k$  be a sequence of distinct candidates not in  $C$ . For each  $i = 1, \dots, k$ , let  $\mathcal{R}_i$  denote a preference profile in which for each  $j = 1, \dots, k$ ,  $j \neq i$ ,  $D^j$  is decloned to  $c^j$ . Then  $\mathcal{R}$  is single-peaked if and only if each of the profiles  $\mathcal{R}_i$ ,  $1 \leq i \leq k$ , is single-peaked.*

*Proof.* The “only if” direction follows from Proposition C.2. For the “if” direction, we will give a proof for the case  $k = 2$ ; the general case follows by induction.

Set  $\mathcal{R}' = \mathcal{R}_1(D^1 \mapsto c^1) = \mathcal{R}_2(D^2 \mapsto c^2)$ , and let  $D = D^1$ ,  $c = c^1$ . By Proposition C.2,  $\mathcal{R}'$  is single-peaked; let  $>'$  be some order that witnesses this. The general idea of our proof is as follows: We will first show that there are orders  $>^1$  and  $>^2$ , both very similar to  $>'$ , witnessing single-peakedness of  $\mathcal{R}_1$  and  $\mathcal{R}_2$ , respectively. We will then show that the “edits” needed to turn  $>'$  into  $>^1$  and the “edits” needed to turn  $>'$  into  $>^2$  are independent and thus we can turn  $>'$  into an order witnessing single-peakedness of  $\mathcal{R}$ .

We will now construct an order  $>^1$  that is very similar to  $>'$  and witnesses the single-peakedness of  $\mathcal{R}_1$ . We start with an arbitrary preference order  $>$  that witnesses the single-peakedness of  $\mathcal{R}_1$ . This order may not have the properties that we are interested in; therefore, we will construct  $>^1$  by taking a “hybrid” of  $>$  and  $>'$ . We will consider two cases.

**$D$  is a clone set of the first type with respect to  $>$ .** We claim that in this case we can construct  $>^1$  from  $>'$  by replacing  $c$  with the members of  $D$ , ranked either according to  $>$  or according to  $\lessdot$ .

Indeed, let  $p_1$  and  $p_2$  be the extreme peaks of  $\mathcal{R}'$  such that  $p_1 >' p_2$ , and let  $B = \{b \mid p_1 >' b >' p_2\}$ . If  $|\text{peak}(\mathcal{R}')| = 1$ , set  $p_1 = p_2$  to be the unique member of  $\text{peak}(\mathcal{R}')$  and let  $B = \{p_1\}$ .

Let  $>^c$  be the order obtained from  $>$  by replacing the occurrence of  $D$  with  $c$ ; if  $p_2 >^c p_1$ , reverse  $>^c$ . The proof of Proposition C.2 shows that  $\mathcal{R}'$  is single-peaked with respect to  $>^c$ . Thus, by Theorem C.1,  $>^c$  and  $>'$  agree on  $B$  and rank the members of  $B$  consecutively.

Suppose first that  $c >^c B$ . If  $c >' B$  then we obtain  $>^1$  from  $>'$  by replacing the occurrence of  $c$  with the members of  $D$ , ranked in the order of  $>$ . If  $B >' c$ , we replacing the occurrence of  $c$  with the members of  $D$ , ranked in the order of  $\lessdot$ . It is easy to see that  $\mathcal{R}_1$  is single-peaked with respect to  $>^1$ . The case  $B >^c c$  can be handled similarly. Finally, suppose that  $c \in B$ . Since  $>'$  and  $>^c$  agree on  $B$  and rank members of  $B$  consecutively, it suffices again to replace the occurrence of  $c$  with the members of  $D$ , ranked in the order of  $>$ . Clearly,  $\mathcal{R}_1$  is single-peaked with respect to the resulting order  $>^1$ .

**$D$  is a clone set of the second type with respect to  $>$ .** By Proposition 5.2, there is a (unique) partition of  $C$  into sets  $A_1, A_2, D_1, D_2, P$  such that  $P \neq \emptyset$ ,  $D_1 \neq \emptyset$ ,  $D_2 \neq \emptyset$ , and  $A_1 > D_1 > P > D_2 > A_2$ . Further, for each  $R_i \in \mathcal{R}_1$  it holds that  $P \succ_i D \succ_i A$ . This implies that each  $R'_i \in \mathcal{R}'$  is of the form  $P \succ'_i c \succ'_i A$ . Thus, both  $P$  and  $P \cup \{c\}$  are clone sets for  $\mathcal{R}'$ . Moreover,  $P \cap \text{peak}(\mathcal{R}') \neq \emptyset$ , so by Corollary 5.3  $P$  is a clone set of the first type with respect to  $>'$ . Thus, we have either  $A'_1 >' c >' P >' A'_2$  or  $A'_1 >' P >' c >' A'_2$  for some  $A'_1, A'_2$  such that  $A'_1 \cup A'_2 = A$ ; assume without loss of generality that  $A'_1 >' c >' P >' A'_2$ . Consider the order  $>^1$  given by  $A'_1 >^1 D_1 >^1 P >^1 D_2 >^1 A'_2$ , which agrees with  $>'$  on  $C \setminus D$  and agrees with  $>$  on  $D$ . Clearly,  $\mathcal{R}_1$  is single-peaked with respect to  $>^1$ .

In both cases, we derive an order  $>^1$  that witnesses the single-peakedness of  $\mathcal{R}_1$  from  $>'$  by either replacing the occurrence of  $c$  with the members of  $D$  (if  $D$  is a clone set of the first type), or replacing  $c$  with some members of  $D$  and inserting the remaining members of  $D$  into a clearly specified position in  $>'$ . Now, recall that  $\mathcal{R}_1(D^1 \mapsto c^1) = \mathcal{R}_2(D^2 \mapsto c^2)$ , and therefore  $>'$  also witnesses the single-peakedness of  $\mathcal{R}_2(D^2 \mapsto c^2)$ . Hence, we can derive an order  $>^2$  that witnesses the single-peakedness of  $\mathcal{R}_2$  from  $>'$  in a similar manner. Crucially, the “edits” required to obtain  $>^1$  from  $>'$  are independent of the edits required to obtain  $>^2$  from  $>'$ . Consequently, if we apply these edits jointly to obtain a new order  $>^*$ , this order witnesses that  $\mathcal{R}$  is single-peaked, which is exactly what we need to prove.  $\square$

**Proposition 5.10.** *Fat sausages and strings of sausages are single-peaked.*

*Proof.* It suffices to check that the profiles constructed in the proof of Proposition B.1 are single-peaked. For the string of sausages, this is immediate: the respective order  $>$  is given by  $1 > \dots > m$ . For the fat sausage with  $m = 2k$ ,  $k \geq 2$ , we can use the order  $x_k > \dots > x_1 > y_1 > \dots > y_k$ . For the fat sausage with  $m = 2k + 1$ ,  $k \geq 2$ , we can use the order  $z > x_k > \dots > x_1 > y_1 > \dots > y_k$ . Finally, for the fat sausage with  $m = 3$ , we can set  $1 > 2 > 3$ .  $\square$

**Proposition 5.11.** *Let  $\mathcal{C}$  be a clone structure over a set of candidates  $C$ . Suppose that for each  $Q$ -node of the PQ-tree decomposition  $T(\mathcal{C})$  of  $\mathcal{C}$  it holds that all children, except possibly the leftmost child and the rightmost child, are labeled with singletons, i.e., elements of  $C$ . Then  $\mathcal{C}$  is a single-peaked clone structure.*

*Proof.* Let us fix  $C$  and  $\mathcal{C}$  as in the statement of the theorem and let  $T = T(\mathcal{C})$  be some PQ-tree decomposition of  $\mathcal{C}$ . We first describe a societal axis  $>$ , and then construct a profile  $\mathcal{R}$  that is single-peaked with respect to  $>$  and satisfies  $\mathcal{C}(\mathcal{R}) = \mathcal{C}$ .

We obtain  $>$  as follows. For every pair of candidates  $c', c'' \in C$ , we set  $c' > c''$  if in the DFS traversal of  $T$ , the node representing  $c'$  is visited before the node representing  $c''$ . Note that  $T$  is an ordered tree and thus the order of the DFS traversal is uniquely determined; intuitively, this is simply the left-to-right order of leaves of  $T$ .

To describe the profile  $\mathcal{R}$ , we need some additional notation. As in Definition 5.6, for every node  $v \in T$ , we set  $C_v = \{c \in C \mid c \text{ is a leaf of } T\text{'s subtree rooted in } v\}$ , and let  $D_v = C \setminus C_v$ . For every node  $v \in T$ , we will introduce several preference orders that rank the candidates in  $C_v$  ahead of those in  $D_v$ . This will ensure that the part of  $\mathcal{C}$  that corresponds to  $v$  is implemented correctly. Since each preference order in  $\mathcal{R}$  has to rank all candidates, we will define for each  $v \in T$  an order  $\succ_v$  on  $D_v$  that will be used for ranking the candidates in  $D_v$  in the preference profiles that correspond to  $v$ .

The order  $\succ_v$  is defined as follows. Let  $P = (v_1, \dots, v_k)$  be the (unique) path from  $v$  to the root, where  $v = v_1$ ,  $v_k$  is the root, and for each  $i = 1, \dots, k - 1$  the node  $v_{i+1}$  is the parent of  $v_i$ . Let  $c, c'$  be two candidates in  $D_v$ . Let  $v_i$  be the first node of  $P$  that lies on the path from  $c$  to the root. Similarly, let  $v_j$  be the first node of  $P$  that lies on the path from  $c'$  to the root. Then  $\succ_v$  orders  $c$  and  $c'$  as follows:

1. If  $i < j$  then  $c \succ_v c'$ .
2. If  $j > i$  then  $c' \succ_v c$ .
3. If  $i = j$  then:
  - (a) If both  $c$  and  $c'$  are to the left of  $C_{v_{i+1}}$  in  $T$ , then  $c \succ_v c'$  if and only if  $c' > c$ .
  - (b) If both  $c$  and  $c'$  are to the right of  $C_{v_{i+1}}$  in  $T$ , then  $c \succ_v c'$  if and only if  $c > c'$ .
  - (c) If  $c$  and  $c'$  are on the opposite sides of  $C_{v_{i+1}}$  in  $T$ , then the one to the left of  $C_{v_{i+1}}$  precedes in  $\succ_v$  the one to the right of  $C_{v_{i+1}}$ .

Let  $\succ$  be some preference order over  $C_v$  that is single-peaked with respect to  $>$ . The reader can verify that any preference order that ranks  $C_v$  above  $D_v$ , agrees with  $\succ$  on  $C_v$ , and agrees with  $\succ_v$  on  $D_v$  is single-peaked with respect to  $>$ . Further, for each clone  $D \in \mathcal{C}$  such that either  $C_v \subseteq D$  or  $C_v \cap D = \emptyset$  it holds that the members of  $D$  are ranked consecutively in  $\succ_v$ . (The last claim uses the fact that for any Q-node only its leftmost and rightmost child can be non-singletons.)

We can now describe the profile  $\mathcal{R}$ . For each node  $v$  we construct several preference orders as follows.

Let  $v$  be a P-node with children  $v_1, \dots, v_k$ . For each  $i = 1, \dots, k$ , we add four preference orders, which we will denote by  $R_{v_i}^1, R_{v_i}^2, R_{v_i}^3$ , and  $R_{v_i}^4$ . Each of them ranks  $C_v$  above  $D_v$  and agrees with  $\succ_v$  on  $D_v$ . Thus, it remains to describe how they order the members of  $C_v$ . Set  $A = C_{v_1} \cup \dots \cup C_{v_{i-1}}$ ,  $B = C_{v_{i+1}} \cup \dots \cup C_{v_k}$ . We have (see the description below for clarification):

$$\begin{aligned} R_{v_i}^1 &: C_{v_i} \succ A \succ B, \\ R_{v_i}^2 &: \overleftarrow{C_{v_i}} \succ A \succ B, \\ R_{v_i}^3 &: C_{v_i} \succ B \succ A, \\ R_{v_i}^4 &: \overleftarrow{C_{v_i}} \succ B \succ A. \end{aligned}$$

For each occurrence of  $A$  and  $B$  in the above preference orders, we order member of  $A$  and  $B$  either following  $>$  or the reverse of  $>$ , whichever way is required to ensure single-peakedness. Each occurrence of  $C_{v_i}$  corresponds to ranking the members of  $C_{v_i}$  according to  $>$ , and each occurrence of  $\overleftarrow{C_{v_i}}$  corresponds to ranking the members of  $C_{v_i}$  according to  $>$ .

The preference orders  $R_{v_i}^j$ ,  $j = 1, \dots, 4$ , ensure that any clone in  $\mathcal{R}$  that contains both a member of  $C_{v_i}$  and a member of  $C_v \setminus C_{v_i}$  has to contain all of  $C_v$ . Further, for each clone  $D \in \mathcal{C}$ , the members of  $D$  are ranked consecutively in each  $R_{v_i}^j$ ,  $j = 1, \dots, 4$ .

Now, let  $v$  be a Q-node in  $T$  with children  $v_1, \dots, v_k$ ; note that  $k \geq 3$ , since in our construction of PQ-trees all nodes with 2 children are labeled as P-nodes. Then we introduce voters  $R_{v_1}^1, R_{v_1}^2, R_{v_1}^3$ , and  $R_{v_1}^4$ , and  $R_{v_k}^1, R_{v_k}^2, R_{v_k}^3$ , and  $R_{v_k}^4$ , defined in the same way as for a P-node. This completes the description of  $\mathcal{R}$ .

Clearly, each set in  $\mathcal{C}$  is a clone in the preference profile  $\mathcal{R}$ . Conversely, no set  $D \in 2^C \setminus \mathcal{C}$  is a clone in  $\mathcal{R}$ . Indeed, fix a subset  $D \in 2^C \setminus \mathcal{C}$ , let  $c$  be the minimal element of  $D$  with respect to  $>$ , and let  $c'$  be the maximal element of  $D$  with respect to  $>$ . Now consider the path  $v_1, \dots, v_t$  from  $c$  to  $c'$  in  $T$ . By construction of  $\mathcal{R}$  we have  $\bigcup_{i=1}^t C_{v_i} \subseteq D$ . However, by the choice of  $c$  and  $c'$  we have  $D = \bigcup_{i=1}^t C_{v_i} \in \mathcal{C}$ , which is a contradiction.  $\square$

## D Complete Decloning Algorithm

In this section we give a polynomial-time algorithm that finds optimal decloning toward a single-peaked profile without making any assumptions on the structure of the initial profile.

We start by proving a preliminary lemma, which may be of independent interest.

**Lemma D.1.** *Let  $\mathcal{R}$  be a single-peaked preference profile over a candidate set  $C$ , and let  $D \in \mathcal{C}(\mathcal{R})$  be a clone set such that the set family  $\mathcal{D} = \{X \in \mathcal{C}(\mathcal{R}) \mid X \subseteq D\}$  is a string of sausages. If  $|D| \geq 3$ , then  $\mathcal{R}$  is single-peaked with respect to some order  $>$  such that  $D$  is a clone set of the first type with respect to  $>$ .*

*Proof.* Suppose that  $|D| \geq 3$  and  $\mathcal{R}$  is single-peaked with respect to some order  $>'$  such that  $D$  is a clone set of the second type with respect to  $>'$ . By Proposition 5.2, there is a partition of  $C$  into sets  $P, D_1, D_2, A_1, A_2$  such that  $D = D_1 \cup D_2$ ,  $D_1 \neq \emptyset$ ,  $D_2 \neq \emptyset$ , and  $A_1 >' D_1 >' P >' D_2 >' A_2$ . By the same proposition,  $\text{peak}(\mathcal{R}) \subseteq P$ .

Assume without loss of generality that the first voter in  $\mathcal{R}$  ranks the candidates in  $D$  as  $d_1 \succ_1 \dots \succ_1 d_m$ . We will now argue that each voter in  $\mathcal{R}$  ranks the candidates in  $D$  according to  $\succ_1$ . Since  $\mathcal{D}$  is a string of sausages, each voter's ranking of the candidates in  $D$  coincides with either  $\succ_1$  or  $\overleftarrow{\succ}_1$ . Now, suppose that some voter  $i$  ranks the candidates in  $D$  according to  $\overleftarrow{\succ}_1$ . Since  $|D| \geq 3$ , at least one of the sets  $D_1, D_2$  has at least two elements. Assume without loss of generality that  $|D_1| \geq 2$  and let  $d, d'$  be two candidates in  $D_1$  such that  $d \succ_1 d'$  (and hence  $d' \succ_i d$ ). Since  $\text{peak}(R_1) \in P$  and  $D_1 > P$ , we have  $d' >' d$ . However, since  $\text{peak}(R_i) \in P$ , it has to be the case that  $d \succ_i d'$ , a contradiction.

This implies that  $\mathcal{R}$  is single-peaked with respect to an order  $>$  that agrees with  $>'$  on  $C \setminus D$ , agrees with  $\succ_1$  on  $D$ , and is of the form  $A_1 > P > D > A_2$ . Clearly,  $D$  is a clone of the first type with respect to  $>$ .  $\square$

We will now use Lemma D.1 to show that it is never beneficial to partially collapse a string of sausages into a clone of size 3 or more.

**Proposition D.2.** *Let  $\mathcal{R}$  be a preference profile over a set of candidates  $C$ , and let  $D \in \mathcal{C}(\mathcal{R})$  be a clone set such that the set family  $\mathcal{D} = \{X \in \mathcal{C}(\mathcal{R}) \mid X \subseteq D\}$  is a string of sausages. Let  $D' \in \mathcal{D}$  be a clone set,  $D' \neq D$ ,  $|D'| \geq 2$ , and let  $d'$  be a candidate not in  $C$ . If  $\mathcal{R}' = \mathcal{R}(D' \mapsto d')$  is single-peaked and  $|D \setminus D' \cup \{d'\}| \geq 3$ , then  $\mathcal{R}$  is single-peaked as well.*

*Proof.* Let  $D = \{d_1, \dots, d_m\}$ . We can assume that the first voter ranks the candidates in  $D$  as  $d_1 \succ_1 \dots \succ_1 d_m$ , and  $D' = \{d_i, \dots, d_j\}$ , where  $1 \leq i < j \leq m$ . Note that all voters in  $\mathcal{R}$  rank the candidates in  $D$  either according to  $\succ_1$  or according to  $\overleftarrow{\succ}_1$ , and therefore  $D \setminus \{d_1, d_m\}$  does not contain any peaks of  $\mathcal{R}$ . Set  $D^* = D \setminus D' \cup \{d'\}$ . Clearly,  $D^*$  is a string of sausages in  $\mathcal{R}'$ . Thus, by Lemma D.1 the profile  $\mathcal{R}'$  is single-peaked with respect to some order  $>'$  such that  $D^*$  is a clone set of the first type with respect to  $>'$ , i.e., we have  $A_1 >' D^* >' A_2$ , where  $A_1 \cup A_2 = C \setminus D$ .

Suppose that  $\text{peak}(\mathcal{R}') \not\subseteq D^*$ . Then, since  $D^*$  is a string of sausages in  $\mathcal{R}'$ , the restriction of  $>'$  on  $D^*$  is either of the form

$$d_1 >' \dots >' d_{i-1} >' d' >' d_{j+1} >' \dots >' d_m$$

or of the form

$$d_m >' \dots >' d_{j+1} >' d' >' d_{i-1} >' \dots >' d_1;$$

by reversing  $>'$  if necessary, we can assume the former. Consider the order  $>$  on  $C$  that agrees with  $>'$  outside of  $D$  and ranks the elements of  $D$  as  $d_1 > \dots > d_m$ . It is easy to see that  $\mathcal{R}$  is single-peaked with respect to  $>$ .

On the other hand, suppose that  $\text{peak}(\mathcal{R}') \subseteq D^*$ . This means that  $\text{peak}(\mathcal{R}) \subseteq \{d_1, d_m\}$ . If  $|\text{peak}(\mathcal{R}')| = 2$  then  $>'$  is of the form as in the paragraph above and we can obtain  $>$  from it the same way. If  $|\text{peak}(\mathcal{R}')| = 1$  then all preference orders in  $\mathcal{R}'$  rank voters in  $D^*$  identically. We obtain  $>$  from  $>'$  by replacing the occurrence of  $D^*$  with the occurrence of  $D$ , orders from  $d_1$  to  $d_m$ .  $\square$

We will now briefly describe our algorithm  $\text{DECLONESP}(\mathcal{R})$  that finds an optimal decloning of  $\mathcal{R}$  towards a single-peaked profile. It proceeds as  $\text{BASICDECLONESP}(\mathcal{R})$ , with one exception. Specifically, when  $\text{DECLONESP}(\mathcal{R})$  processes a node  $v$  of type Q with  $m$  children  $v_1, \dots, v_m$  and discovers that  $v$  cannot be colored white, it considers two ways of splitting its children into two contiguous groups, namely,  $(\{v_1\}, \{v_2, \dots, v_m\})$  and  $(\{v_1, \dots, v_{m-1}\}, \{v_m\})$ . For each split, it declones the non-singleton group, i.e., removes the respective branches of the tree and replaces them with one black node. It then checks if the resulting profile is single-peaked. If yes, it might be possible to recursively expand the singleton node, so  $\text{DECLONESP}(\mathcal{R})$  calls itself recursively on the respective subtree. After the recursive calls that correspond to both splits return, the algorithm chooses the better split. We postpone the formal description of the algorithm and its proof of correctness to the full version of the paper; in essence, Proposition D.2 shows that it suffices to consider the splits of the form described above, and an argument similar to that in the proof of Theorem 5.8 shows that different branches of the tree can be handled independently.

## E Material Missing from Section 6

**Proposition 6.2.** *The problem of checking if a given profile is single-crossing is in P.*

*Proof.* Suppose that we are given a preference profile  $\mathcal{R} = (R_1, \dots, R_n)$  over a set of candidates  $C = \{c_1, \dots, c_m\}$ . For each  $i = 1, \dots, n$ , we will check if there exists an order  $\triangleleft$  over  $[n]$  in which voter  $i$  appears first; we return “yes” if the answer is positive for some  $i$ .

Without loss of generality, we can focus on the case  $i = 1$  and assume that voter 1 ranks the candidates as  $c_1 \succ_1 \dots \succ_1 c_m$ . Now, consider a directed graph  $G$  with a vertex set  $[n]$  that has an edge from  $j$  to  $k$  if there exists a pair of candidates  $(c_x, c_y)$  with  $x < y$  such that  $c_x \succ_j c_y$ , but  $c_y \succ_k c_x$ . Clearly, for every pair of nodes  $(j, k)$  at least one of the edges  $(j, k)$  and  $(k, j)$  is present in this graph. The single-crossing condition immediately implies that if there is an edge from  $j$  to  $k$ , then  $j$  precedes  $k$  in  $\triangleleft$ . Therefore, if  $G$  has a cycle,  $\mathcal{R}$  is not single-crossing with respect to any ordering of voters in which voter 1 appears first. On the other hand, if  $G$  is acyclic, it induces a total order over  $[n]$ , and it is immediate that  $\mathcal{R}$  is single-crossing with respect to this order.  $\square$

We will now construct the tools needed to prove Theorems 6.3 and 6.4. The following is an immediate but useful corollary to the definition of single-crossing profiles; to some

degree, it justifies viewing single-crossing profiles as collections of voters over some spectrum of views.

**Observation E.1.** *If profile  $\mathcal{R} = (R_1, \dots, R_n)$  is single-crossing with respect to the order  $1 \triangleleft 2 \triangleleft \dots \triangleleft n$  then  $\mathcal{R}' = (R_1, \dots, R_n, \overleftarrow{R_1})$  is single-crossing as well.*

We now define a family of single-crossing profiles that implements a fat sausage; we will also use it within the reduction in Theorem 6.4.

**Definition E.2.** *Let  $m$  be a positive integer,  $m > 2$ , and let  $C$  be an  $m$ -element candidate set. Rename the candidates so that  $C = \{a, b, c_1, \dots, c_{m-2}\}$ . We call each profile  $\mathcal{R} = \{R_1, \dots, R_m\}$  of the form:*

$$\begin{aligned} R_1 : & a \succ_1 b \succ_1 c_2 \succ_1 \dots \succ_1 c_{m-2} \\ R_2 : & b \succ_2 a \succ_2 c_2 \succ_2 \dots \succ_1 c_{m-2} \\ R_3 : & b \succ_3 c_2 \succ_3 a \succ_3 \dots \succ_1 c_{m-2} \\ & \vdots \\ R_m : & b \succ_m c_2 \succ_m c_3 \succ_m \dots \succ_m c_{m-2} \succ_m a \end{aligned}$$

a slide over  $C$ . We refer to  $a$  as the sliding candidate.

**Proposition E.3.** *Let  $C$  be a set of candidates and let  $\mathcal{R}$  be a slide over  $C$ .  $\mathcal{R}$  is single-crossing and  $\mathcal{C}(\mathcal{R})$  is a fat sausage.*

*Proof.* Let  $\mathcal{R} = (R_1, \dots, R_m)$  be a slide over  $C$  (we will use the notation as in Definition E.2).  $\mathcal{R}$  is clearly single-crossing with respect to the order  $1 \triangleleft \dots \triangleleft m$ . We claim that  $\mathcal{C}(\mathcal{R})$  is a fat sausage. To see this, first note that if  $D$  is a clone set and  $a, b \in D$  then  $D = C$ . This is so, because  $R_m$  ranks  $b$  first and  $a$  last. Let  $D$  be a clone set in  $\mathcal{C}(\mathcal{R})$ . It is easy to verify that if  $D$  contains more than one candidate then it must contain  $a$  and  $b$ , and thus  $D = C$ . This completes the proof.  $\square$

**Proposition E.4.** *Let  $m$  be a positive integer,  $m > 2$ , and let  $\mathcal{R} = (R_1, \dots, R_m)$  be a slide over  $m$ -candidate set  $C$ .  $\mathcal{R}$  is not single-crossing with respect to orders other than  $1 \triangleleft_1 2 \triangleleft_1 \dots \triangleleft_1 m$  and  $m \triangleleft_2 m-1 \triangleleft_2 \dots \triangleleft_2 1$ .*

*Proof.* For the sake of contradiction assume that there is an order  $\triangleleft$  different from  $\triangleleft_1$  and  $\triangleleft_2$  with respect to which  $\mathcal{R}$  is single-crossing. Suppose first that there are some  $i, j, k \in [m]$  such that  $i < j < k$  and  $i \triangleleft k \triangleleft j$ .

Let  $a$  be the sliding candidate in  $\mathcal{R}$ . Since  $j \neq m$ , there is some candidate  $c$  such that both  $R_i$  and  $R_j$  rank  $a$  above  $c$ , but  $R_k$  ranks  $c$  above  $a$ . However, this immediately implies that  $\mathcal{R}$  is not single-crossing with respect to  $\triangleleft$ . The remaining three cases (in which there exist  $i < j < k$  such that  $j \triangleleft i \triangleleft k$ ,  $j \triangleleft k \triangleleft i$ , or  $k \triangleleft i \triangleleft j$ ) can be handled similarly.  $\square$

**Theorem 6.3.** *For every clone structure  $\mathcal{C}$  there exists a single-crossing profile  $\mathcal{R}$  such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$ .*

*Proof.* It is easy to see that each string of sausages can be implemented with a single-crossing profile because a profile with a single voter suffices. Fat sausages can be implemented by Proposition E.3. These are the only two types of irreducible clone structures and, thus, to prove the theorem it remains to show that clone structures implementable using single-crossing profiles are closed under embeddings.

Let  $\mathcal{C}$  and  $\mathcal{D}$  be two clone structures over disjoint sets  $C$  and  $D$ , such that both  $\mathcal{C}$  and  $\mathcal{D}$  can be implemented with a single-crossing profile. Let  $\mathcal{R} = (R_1, \dots, R_n)$  be a single-crossing profile over  $C$  such that  $\mathcal{C} = \mathcal{C}(\mathcal{R})$  and let  $\mathcal{Q} = (Q_1, \dots, Q_n)$  be a single-crossing profile over  $D$  such that  $\mathcal{D} = \mathcal{C}(\mathcal{Q})$ . Note that since we can freely duplicate preference orders in a single-crossing profile, we can assume that both  $\mathcal{R}$  and  $\mathcal{Q}$  have the same number of voters. Further, by Corollary E.1, we can assume that  $Q_1$  and  $Q_n$  are reverses of each other, and we can assume that both  $\mathcal{R}$  and  $\mathcal{Q}$  are single-crossing with respect to the standard order over integers  $(1 \triangleleft 2 \triangleleft \dots \triangleleft n)$ .

Fix an arbitrary candidate  $c \in C$ . We will construct a profile  $\mathcal{P} = (P_1, \dots, P_{2n-1})$  such that  $\mathcal{C}(c \rightarrow \mathcal{D}) = \mathcal{C}(\mathcal{P})$ . For each  $i = 1, \dots, n$ , we define  $P_i$  to be identical to  $R_i$ , except that we replace the occurrence of  $c$  with  $Q_1$ . For each  $k = 2, \dots, n$ , we define  $P_{(n-1)+k}$  to be identical to  $R_n$ , except that we replace the occurrence of  $c$  with  $Q_k$ . (Somewhat abusing our notation, we could write  $\mathcal{P} = (R_1(c \rightarrow Q_1), R_2(c \rightarrow Q_1), \dots, R_n(c \rightarrow Q_1), R_n(c \rightarrow Q_2), \dots, R_n(c \rightarrow Q_n))$ .)

It is easy to verify that since both  $\mathcal{R}$  and  $\mathcal{Q}$  are single-crossing then so is  $\mathcal{P}$ . Similarly, it is easy to verify that  $\mathcal{D} \subseteq \mathcal{C}(\mathcal{P})$  and that for each  $A \in \mathcal{C}$ , if  $c \notin A$ , then  $A \in \mathcal{C}(\mathcal{P})$  and if  $c \in A$  then  $(A \setminus \{c\}) \cup D \in \mathcal{C}(\mathcal{P})$ . Thus, it remains to show that  $\mathcal{P}$  does not contain any “parasite” clones.

Clearly, any clone set in  $\mathcal{C}(\mathcal{P})$  that contains members of  $C$  only or members of  $D$  only belongs to  $\mathcal{C}(c \rightarrow \mathcal{D})$ . Consider preference orders  $P_n$  and  $P_{2n-1}$ . By construction,  $P_n$  is  $R_n$  with  $c$  replaced by  $Q_1$  and  $P_{2n-1}$  is  $R_n$  with  $c$  replaced by  $\overleftarrow{Q_1}$  (recall the assumption that  $Q_n = \overleftarrow{Q_1}$ ). Thus, any clone set of  $\mathcal{C}(\mathcal{P})$  that contains both a member of  $C$  and a member of  $D$  must contain all members of  $D$ . By construction, any clone set  $X \in \mathcal{C}(\mathcal{P})$  that contains all members of  $D$  belongs to  $\mathcal{C}(c \rightarrow \mathcal{D})$ . This completes the proof.  $\square$

The above embedding construction stands in a sharp contrast to the construction used for unrestricted profiles (Proposition 3.11). There, we could embed one clone structure into the other without increasing the number of voters needed to implement the profiles. Here, every embedding operation nearly doubles the number of votes. Thus, for unrestricted preferences we need between 1 and 3 voters to implement any given clone structure, but for single-crossing profiles our construction may require a number of voters that is exponential in  $|C|$ . On the other hand, our construction is certainly not optimal (to see this, compare the clone structure built in the proof of Theorem 6.4 and the profile that implements it there, and the profile that would arise from applying the construction from Theorem 6.3 to implement this clone structure). It is interesting to ask if it is possible to implement every clone structure over candidate set  $C$  with a single-crossing profile with at most  $\text{poly}(|C|)$  preference orders.



Let us now turn to the issue of decloning toward a single-crossing profile. Unfortunately, as opposed to the case of single-peaked elections, for single-crossing the problem is NP-complete. Our reduction uses the standard NP-complete problem Exact Cover by 3-Sets (X3C).

**Definition E.5.** An instance  $I = (B, \mathcal{S})$  consists of a base set  $B = \{b_1, \dots, b_{3k}\}$  and a collection  $\mathcal{S} = \{S_1, \dots, S_s\}$  of 3-element subsets of  $B$ . It is a yes-instance if there exists a set  $A \subseteq \{1, \dots, s\}$  such that (a)  $\bigcup_{i \in A} S_i = B$  and (b) for each  $i, j \in A$ ,  $S_i \cap S_j = \emptyset$ .

**Theorem 6.4.** Given a profile  $\mathcal{R}$  over a candidate set  $C$  and a positive integer  $k$ , it is NP-complete to decide if there exists a single-crossing profile  $\mathcal{R}'$  with  $c(\mathcal{R}') \geq k$  such that  $\mathcal{R}'$  can be obtained from  $\mathcal{R}$  by decloning.

*Proof.* Proposition 6.2 implies that this problem is in NP: it suffices to guess clone sets that need to be decloned and use the algorithm from the proof of Proposition 6.2 to verify that the resulting preference profile is single-crossing.

To prove that this problem is NP-hard, we give a reduction from X3C. Let  $I = (B, \mathcal{S})$  be an input instance of X3C with  $B = \{b_1, \dots, b_{3k}\}$  and  $\mathcal{S} = \{S_1, \dots, S_s\}$ . By duplicating sets in  $\mathcal{S}$  if necessary, we can assume that  $s > 3k$ .

We construct a profile  $\mathcal{R}$  in stages. First let  $\mathcal{P} = (P_1, \dots, P_s)$  be a slide over  $[s]$ ; to be specific, we pick 1 to be the sliding candidate, but this choice is irrelevant for the proof. For each  $i = 1, \dots, s$ , we set  $\mathcal{P}'_i$  to be a group of  $2s$  preference orders, denoted  $P'_{i,1}, \dots, P'_{i,t}$ , each identical to  $P_i$ ; further in our construction we will modify the members of each group appropriately. We define  $\mathcal{P}'$  to consist exactly of these  $s$  groups of orders: that is, abusing notation,  $\mathcal{P}' = \mathcal{P}'_1 + \dots + \mathcal{P}'_s$ . Intuitively, for  $i = 1, \dots, 3k$ , group  $\mathcal{P}'_i$  corresponds to the element  $b_i$  of  $B$  and, after further modifications that we will introduce in the profile, the role of group  $\mathcal{P}'_i$  will be to ensure that it is impossible to pick two sets from  $\mathcal{S}$  that both contain  $b_i$ . The remaining  $s - 3k$  groups are added for the sake of uniformity and to maintain the slide structure within  $\mathcal{P}'$ .

Let  $\{C_1, \dots, C_s\}$  be a family of disjoint candidate sets such that  $|C_j| = 6s$  for each  $j = 1, \dots, s$ . For each  $j = 1, \dots, s$ , we let  $\mathcal{Q}^j = (Q_1^j, \dots, Q_{6s}^j)$  be a slide over  $C_j$  (picking an arbitrary member of  $C_j$  to be the sliding candidate). We obtain profile  $\mathcal{Q}^j$  by splitting  $\mathcal{Q}^j$  into three contiguous groups of size  $2s$  each and swapping the  $2j$ -th and  $(2j - 1)$ -th voter in each group (that is,  $2j$  and  $2j - 1$ ,  $2j + 2s$  and  $2j - 1 + 2s$ , and  $2j + 4s$  and  $2j - 1 + 4s$ ). Clearly,  $\mathcal{Q}^j$  is single-crossing with respect to the voter ordering  $\triangleleft_j$  obtained from the standard ordering  $1 \triangleleft \dots \triangleleft 6s$  by swapping the same pairs of voters, namely,  $2j$  and  $2j - 1$ ,  $2j + 2s$  and  $2j - 1 + 2s$ , and  $2j + 4s$  and  $2j - 1 + 4s$ . Moreover, by Proposition E.4 the profile  $\mathcal{Q}^j$  is single-crossing with respect to  $\triangleleft_j$  and its reverse, but not with respect to any other order.

We construct our final profile  $\mathcal{R}$  from  $\mathcal{P}'$  by embedding the profiles  $\mathcal{Q}^j$ ,  $j = 1, \dots, 6s$ , into it. Specifically, for each  $S_j \in \mathcal{S}$  such that  $S_j = \{b_x, b_y, b_z\}$ ,  $x < y < z$ , we replace candidate  $j$  with one of the preference orders from  $\mathcal{Q}^j = (Q_1^j, \dots, Q_{6s}^j)$  as follows:

1. For each  $i$ ,  $1 \leq i < x$ , in group  $\mathcal{P}'_i$  we replace candidate  $j$  with preference order  $Q_1^j$ .

2. For  $i = x$  and for each  $\ell = 1, \dots, 2s$ , we replace candidate  $j$  in  $P'_{i,\ell}$  with  $Q_\ell^j$ .
3. For each  $i$ ,  $x < i < y$ , in group  $\mathcal{P}'_i$  we replace candidate  $j$  with preference order  $Q_{2s}^j$ .
4. For  $i = y$  and for each  $\ell = 1, \dots, 2s$ , we replace candidate  $j$  in  $P'_{i,\ell}$  with  $Q_{2s+\ell}^j$ .
5. For each  $i$ ,  $y < i < z$ , in group  $\mathcal{P}'_i$  we replace candidate  $j$  with preference order  $Q_{4s}^j$ .
6. For  $i = z$  and for each  $\ell = 1, \dots, t$ , we replace candidate  $j$  in  $P'_{i,\ell}$  with  $Q_{4s+\ell}^j$ .
7. For each  $i$ ,  $z < i < s$ , in group  $\mathcal{P}'_i$  we replace candidate  $j$  with preference order  $Q_{6s}^j$ .

As a result, we obtain a profile  $\mathcal{R}$  over the candidate set  $C = \bigcup_{i=1}^s C_i$ , consisting of  $s$  groups of voters,  $\mathcal{R}_1, \dots, \mathcal{R}_s$ , where each group  $\mathcal{R}_i$  contains  $2s$  voters denoted by  $R_{i,1}, \dots, R_{i,2s}$ . Clearly,  $\mathcal{R}$  can be constructed in time polynomial in  $s$ . We claim that  $I$  is a *yes*-instance of X3C if and only if there exists a single-crossing profile  $\mathcal{T}$  with  $c(\mathcal{T}) \geq 2sk + (s - k)$  that can be obtained by decloning  $\mathcal{R}$ .

Let us assume that there exists a single-crossing profile  $\mathcal{T}$  with  $c(\mathcal{T}) \geq 2sk + (s - k)$  that can be obtained by decloning  $\mathcal{R}$ . Just as  $\mathcal{R}$  was divided into  $s$  groups  $\mathcal{R}_1, \dots, \mathcal{R}_s$ ,  $\mathcal{T}$  is divided into corresponding  $s$  groups  $\mathcal{T}_1, \dots, \mathcal{T}_s$ . We will show that in this case  $I$  is a *yes*-instance of X3C.

Note that  $\mathcal{C}(\mathcal{R})$  is a composition of a fat sausage over  $[s]$  and fat sausages over the candidate sets  $C_i$  for  $i = 1, \dots, s$ . Thus,  $C_1, \dots, C_s$  are the only nontrivial clones in  $\mathcal{C}(\mathcal{R})$ . Define  $A = \{i \mid C_i \text{ is not decloned in } \mathcal{T}\}$ . We claim that  $A$  corresponds to an exact cover of  $B$ , i.e.,  $\bigcup_{i \in A} S_i = B$  and the sets  $S_i$ ,  $i \in A$ , are pairwise disjoint.

Since  $c(\mathcal{T}) \geq 2sk + (s - k)$  and  $C_i$ s are the only nontrivial clones in  $\mathcal{T}$ , it must be the case that  $|A| \geq k$ . Now, consider two sets  $S_j$  and  $S_k$  such that  $j, k \in A$ . Assume that there exists an element  $b_i$  such that  $b_i \in S_j \cap S_k$ . We will show that this implies that  $\mathcal{T}$  is not single-crossing. Indeed, since  $C_j$  is not decloned and  $b_i \in S_j$ , any order of voters witnessing single-crossingness of  $\mathcal{T}$  has to order the voters in  $\mathcal{T}_i = (T_{i,1}, \dots, T_{i,t})$  according to  $\triangleleft_j$  or its reverse. Similarly, since  $b_i \in S_k$  and  $C_k$  has not been decloned, any order witnessing single-crossingness of  $\mathcal{T}$  has to order the voters in  $\mathcal{T}_i$  according to  $\triangleleft_k$  or its reverse. By construction,  $\triangleleft_j$  and  $\triangleleft_k$  are neither identical nor each others' reverses. Thus, it must be the case that  $S_j \cap S_k = \emptyset$ . In consequence, it must be the case that  $|A| = k$  and  $\bigcup_{i \in A} S_i = B$ .

For the other direction, let us assume that  $I$  is a *yes*-instance and let  $A \subseteq \{1, \dots, s\}$  be such that  $\bigcup_{i \in A} S_i = B$  and for each  $i, j \in A$  we have  $S_i \cap S_j = \emptyset$ . Let  $\mathcal{T}$  be a profile obtained from  $\mathcal{R}$  by decloning each set  $C_i$  such that  $i \notin A$ . We claim that  $\mathcal{T}$  is single-crossing. To see this, it suffices to take an order  $\triangleleft$  that orders the groups  $\mathcal{T}_1, \dots, \mathcal{T}_n$  of  $\mathcal{T}$  as  $\mathcal{T}_1 \triangleleft \mathcal{T}_2 \triangleleft \dots \triangleleft \mathcal{T}_s$ , within each group  $\mathcal{T}_i$ ,  $i = 1, \dots, 3k$ , orders the voters according to  $\triangleleft_j$ , where  $j \in A$  and  $b_i \in S_j$  (by choice of  $A$  such a  $j$  is unique), and within each group  $\mathcal{T}_i$ ,  $i = 3k + 1, \dots, s$ , orders the voters arbitrarily.  $\square$

**Proposition 6.5.** *Given a profile  $\mathcal{R}$  over candidate set  $C$ , a positive integer  $k$ , and an order  $\triangleleft$ , we can decide in polynomial time if there exists a profile  $\mathcal{R}'$  with  $c(\mathcal{R}') \geq k$  that is single-crossing with respect to  $\triangleleft$  and can be obtained from  $\mathcal{R}$  by decloning.*

*Proof.* We can assume without loss of generality that  $\mathcal{R} = (R_1, \dots, R_n)$  is single-crossing with respect to the standard order  $1 \triangleleft \dots \triangleleft n$  over  $[n]$ . Consider any pair of candidates  $(c_x, c_y)$  that violates the single-crossing condition with respect to  $\triangleleft$ , i.e.,  $c_x \succ_1 c_y$ ,  $c_x \succ_n c_y$  and  $c_y \succ_i c_x$  for some  $i$ ,  $1 < i < n$ ; we will write  $x \perp y$  if this is the case. Clearly, if  $\mathcal{R}$  is single-crossing,  $c_x$  and  $c_y$  have to be decloned into the same candidate. Let  $C(x, y)$  be the unique minimal (with respect to set inclusion) subset of candidates such that: (a)  $c_x \in C(x, y)$ ; (b)  $c_y \in C(x, y)$ ; and (c) for every pair of candidates  $c_t, c_z \in C(x, y)$ , and every candidate  $c_w \in C$  such that  $c_t \succ_i c_w \succ_i c_z$  for some  $i \in [n]$  it holds that  $c_w \in C(x, y)$ . The set  $C(x, y)$  is well-defined and can be constructed inductively: we start with  $c_x$  and  $c_y$ , and at each step we add all candidates that appear between some candidates already in  $C(x, y)$  in at least one preference order. A simple inductive argument shows that  $C(x, y)$  is a clone set, and, for  $\mathcal{R}'$  to be single-crossing, all elements of  $C(x, y)$  must be decloned into the same candidate. We further observe that the set family  $\{C(x, y) \mid x \perp y\}$  is laminar, i.e., for every  $x, y, z, t \in [m]$  such that  $x \perp y$  and  $z \perp t$  we have  $C(x, y) \subset C(z, t)$  or  $C(z, t) \subset C(x, y)$  or  $C(x, y) = C(z, t)$ . Thus, an optimal decloning of  $\mathcal{R}$  can be obtained by decloning the maximal (with respect to set inclusion) sets in  $\{C(x, y) \mid x \perp y\}$ .  $\square$